
(19) **KOREAN INTELLECTUAL PROPERTY OFFICE**

KOREAN PATENT ABSTRACTS

(11)Publication number: **1020010109900**
(43)Date of publication of application: **12.12.2001** **A**

(21)Application number:	1020000030638	(71)Applicant:	SAMSUNG ELECTRONICS CO., LTD.
(22)Date of filing:	03.06.2000	(72)Inventor:	CHOI, HO GYU KIM, MIN GU KIM, YUN SEON PARK, JIN SU

(51)Int. Cl **H04B 1/69**

(54) FLEXIBLE DATA RATE MATCHING METHOD AND APPARATUS FOR DATA COMMUNICATION SYSTEM

(57) Abstract:

PURPOSE: A flexible data rate matching method and apparatus for a data communication system are provided to flexibly transmit a data according to a transfer rate without degradation of a performance by uniformly distributing a hole pattern or a repetition pattern in a frame by controlling a set initial value. CONSTITUTION: First parameters(N,L,M,P) required for an FDRT are initialized(401). Parameters D1 and P1 are computed according to a formula given to an algorithm(402). Parameters D2 and P2 are computed according to a formula given to an algorithm(403). When parameters are all determined, k performs counting sequentially performed from 1 to LM. It is checked whether k is multiple of D1 or multiple of D2+1(405). If k is checked to be multiple of D1 or multiple of D2+1, a corresponding kth code word symbol is holed(406). If k is not checked, the value k is increased by 1 (407). It is checked whether k is all counted up to LM and if still there is k, the above operations are repeatedly performed until k=LM+1 is checked(408).

copyright KIPO 2002

Legal Status

Date of request for an examination (20050603)

Notification date of refusal decision (00000000)

Final disposal of an application (application)

Date of final disposal of an application (00000000)

THIS PAGE BLANK (USPTO)

Patent registration number ()

Date of registration (00000000)

Number of opposition against the grant of a patent ()

Date of opposition against the grant of a patent (00000000)

Number of trial against decision to refuse ()

Date of requesting trial against decision to refuse ()

(19) 대한민국특허청 (KR)
(12) 공개특허공보 (A)

(51) 。 Int. Cl. ⁷
H04B 1/69

(11) 공개번호 특2001 - 0109900
(43) 공개일자 2001년12월12일

(21) 출원번호 10 - 2000 - 0030638
(22) 출원일자 2000년06월03일

(71) 출원인 삼성전자 주식회사
윤종용
경기 수원시 팔달구 매탄3동 416

(72) 발명자 박진수
서울특별시서초구반포4동70 - 1한신서래아파트3 - 608
김만구
경기도수원시팔달구영통동973 - 3우성아파트822 - 406
김윤선
서울특별시강남구대치2동쌍용아파트6동607호
최호규
서울특별시서초구잠원동56 - 2신반포2차351 - 603

(74) 대리인 이견주

심사청구 : 없음

(54) 데이터 통신 시스템을 위한 가변 데이터 전송을 정합 방법 및 장치

요약

데이터 통신시스템에서 다양한 전송률을 가변적으로 선택하여 제공하는 가변데이터 전송을 정합 방법 및 장치가 개시되어 있다. 이러한 본 발명에 따르면, 하나의 프레임내에 포함되는 L개의 코드심볼들이 M배로 반복됨에 따라 생성되는 LM개의 심볼들중에서 인터리버 크기에 일치하도록 P개의 심볼들을 천공될 심볼로 결정하는 방법은, 각 행들이 L개의 코드심볼들로 구성되는 M개의 행들중에서 1개의 행을 천공을 행할 행으로 선택하는 과정과, 상기 선택된 행에 포함되는 L개의 코드심볼들중에서 균일하게 분포하는 P개의 심볼들을 천공 심볼들로 결정하는 과정으로 이루어진다. 이때 천공 심볼들은 상기 프레임내의 코드심볼들중 D1 및 D2 위치에 해당하는 심볼들로서 결정된다. 여기서, $D_1 = \lceil L/P \rceil$ 이고, $D_2 = \lceil P_1/P_2 \rceil$ 이고, L은 부호화기 출력 코드심볼의 수이고, M=[NL] 은반복 배수이고, P=LM-λ는 천공될 심볼의 수이고, $P_1 = \lceil LD_1 \rceil$ 는 첫 번째 천공 주기(D1)로 천공될 심볼의 수이고, $P_2 = P - P_1$ 는 두 번째 천공 주기(D2)로 천공될 심볼의 수이다.

대표도
도 12

세일어
가변데이터 전송율, 부호어심볼, 천공패턴

명세서

도면의 간단한 설명

- 도 1은 일반적인 고정형 데이터전송율 전송 장치의 구성을 도시하는 도면.
- 도 2는 도 1에 도시된 장치에 의한 부호어 심볼 프레임 구조를 도시하는 도면.
- 도 3은 종래 기술에 따른 가변형 데이터전송율 전송 장치의 구성을 도시하는 도면.
- 도 4는 도 3에 도시된 장치에 의한 부호어 심볼 프레임 구조를 도시하는 도면.
- 도 5는 도 3에 도시된 장치에 의해 부호어 심볼이 천공되는 패턴의 일 예를 도시하는 도면.
- 도 6은 종래 기술에 따른 가변형 데이터전송 방식이 가지는 문제점을 설명하기 위한 도면.
- 도 7은 본 발명에 따른 가변형 데이터전송 방식에 의해 부호어 심볼이 천공되는 패턴의 일 예를 도시하는 도면.
- 도 8은 본 발명에 따른 가변형 데이터전송 방식에 의해 얻어되는 효과를 설명하기 위한 도면.
- 도 9는 본 발명의 제1실시예에 따른 가변형 데이터전송 방식의 처리흐름을 도시하는 도면.
- 도 10은 본 발명의 제1실시예에 따른 가변형 데이터전송 장치의 일 예를 도시하는 도면.
- 도 11은 본 발명의 제1실시예에 따른 가변형 데이터전송 장치의 다른 예를 도시하는 도면.
- 도 12는 본 발명의 제2실시예 및 제3실시예에 따른 가변형 데이터전송 방식을 설명하기 위한 도면.
- 도 13은 본 발명의 제2실시예에 따른 가변형 데이터전송 방식의 처리흐름을 도시하는 도면.
- 도 14는 본 발명의 제2실시예에 따른 가변형 데이터전송 장치의 일 예를 도시하는 도면.
- 도 15는 본 발명의 제2실시예에 따른 가변형 데이터전송 장치의 다른 예를 도시하는 도면.
- 도 16은 본 발명의 제3실시예에 따른 가변형 데이터전송 방식의 처리흐름을 도시하는 도면.
- 도 17은 본 발명의 제3실시예에 따른 가변형 데이터전송 장치의 일 예를 도시하는 도면.
- 도 18은 본 발명의 제3실시예에 따른 가변형 데이터전송 장치의 다른 예를 도시하는 도면.
- 도 19는 본 발명의 제1실시예에 따른 가변형 데이터전송 방식에 의한 시뮬레이션 결과의 일 예를 도시하는 그래프.
- 도 20은 본 발명의 제1실시예에 따른 가변형 데이터전송 방식에 의한 시뮬레이션 결과의 다른 예를 도시하는 그래프.

발명의 상세한 설명

발명의 목적

발명이 속하는 기술 및 그 분야의 종래 기술

본 발명은 데이터 통신시스템에 관한 것으로, 특히 다양한 전송율을 가변적으로 선택하여 제공하는 가변데이터 전송율 정합 방법 및 장치에 관한 것이다.

일반적으로 위성시스템, 종합정보통신망(ISDN: Intergrated Service Digital Network), 디지털 셀룰라(Digital cellular) 시스템, 광역 부호분할다중접속(W-CDMA: Wide band Code Division Multiple Access) 시스템, UMTS(Universal Mobile Telecommunications System), IMT(International Mobile Telecommunications) - 2000 시스템과 같은 무선 통신시스템에서 채널부호화 방식으로는 주로 컨볼루션 부호(convolutional code)와 단일복호기가 사용되는 선형블록부호 등이 사용되었다. 이러한 채널부호화 방식에 의해 부호화된 심볼들은 채널 인터리버에 의해 인터리빙되는 것이 일반적이다.

전형적인 채널 인터리버는 프레임당 인터리버 크기와 동일한 부호화 심볼의 수를 가지는 프레임을 입력하여 인터리빙하는 형태이었다. 이와 달리 최근의 채널 인터리버는 프레임당 인터리버 크기와 다른 부호화 심볼의 수를 가지는 프레임을 입력하여 인터리빙하는 소위 가변 데이터전송율 전송(Flexible Data Rate Transmission: 이하 "FDRT"라 칭함)방식에 따른 인터리빙 동작을 수행한다.

도 1은 인터리버 크기와 동일한 부호화 심볼의 수를 가지는 프레임을 입력하여 인터리빙하는 비가변형(혹은 고정형) 데이터전송율 전송 방식(Non Flexible Data Rate Transmission)에 따른 채널 인터리버를 도시하는 도면이다.

상기 도 1을 참조하면, 비가변형 데이터전송율 전송(Non FDRT) 방식에서는 채널의 전송율이 고정된 경우 채널 인터리버(channel interleaver) 100에 입력되는 단위 프레임 당 부호화 심볼(Coded symbol)의 수인 L은 항상 인터리버 크기 N과 같은 크기를 가지고 일정하였다. 즉, IS-2000에서 사용되는 RC(Radio configuration) IS-2000에는 RC1, RC2, RC3, RC4, RC5, RC6, RC7, RC8, RC9 등의 다양한 종류의 전송채널을 사용하고 있으며 각각의 경우 데이터 프레임의 크기, 부호율, 인터리빙방식 등의 차이를 가진다. 이에 따라서 제공되는 이미 결정된 일정한 데이터 레이트만이 전송율로 사용되었다.

그러나 FDRT를 사용하는 경우에는 도 3에 도시된 바와 같이 채널 인터리버 100에 입력되는 부호화 심볼의 수가 채널 인터리버 100과 마찬가지로 N으로 동일하나 부호화 심볼 프레임 내의 부호화 심볼의 구성은 입력데이터 레이트 R_s 에 따라서 큰 차이를 보인다.

도 2는 비가변형 데이터전송율 전송(Non FDRT) 방식에 고속데이터 전송율 전송방식의 부호화 심볼 프레임 구조를 도시하는 도면이다.

상기 도 2를 참조하면, 현재 물리채널(Physical channel)이 RC3의 데이터 전송율(Data Rate) = 19.2kbps로 설정되었다고 가정하면, 이때 채널 인터리버 100의 크기 N은 1536이 된다. 만일 사용자가 데이터를 데이터 전송율 20kbps로 전송하고자 한다면, 비가변형 데이터전송율 전송 방식을 사용하는 경우 데이터 전송율을 38.4kbps로 상향조절토록 기지국과 단말기가 초기 협상(Negotiation)과정에서 결정한다. 왜냐하면 20kbps 보다 큰 최소의 데이터 전송율이 38.4kbps이기 때문이다.

따라서 채널 인터리버 100의 크기는 $N=3072(=2 \times 1536)$ 로 2배 증가된다. 채널 부호기(Channel encoder)(도시하지 않음)에 입력되는 입력 데이터 심볼의 구성도 $20\text{kbps} \times 20\text{msec}$ 이외의 부분에 해당되는 빈 구간에는 널 데이터(Null data)가 상위 계층(layer)에 의해 쓰여진다. 따라서 N심볼의 채널 인터리버 크기 중에서 $(38.4 - 20)/38.4 = 47.9$

2%가 널 데이터로 쓰여지고 전송된다. 따라서 수신 심볼에너지 E_s 측면에서는 47.92%의 에너지가 손실되고 있다고 볼 수 있다. 이와 같은 손실이 발생하는 이유는 현재 비가변형 데이터 전송을 전송 방식의 구조에서는 물리계층(Physical layer)에서 이러한 널 데이터를 처리할 수 있는 방법이 없기 때문이다. 만일 널 데이터를 심볼 반복(repetition)으로 사용하려고 해도 현재 F-SCH 구조에서는 심볼 결합(symbol combining)을 할 수 있는 방법이 없다. 또한 입력 데이터 전송율에 따라서 매번 다르므로 이를 상위 계층에서 반드시 사전 전달해야 한다.

실제로 채널 복호기(Channel decoder)를 통과하기 전에 이러한 에너지 복원이 이루어져야 하며 채널 복호기 이후에는 복호된 정보 심볼(Information symbol)만을 가지고 $L1/L2$ 의 상위 계층이 처리하게 되므로 다른 성능개선 방안이 없다.

이러한 비가변형 데이터 전송을 전송방식의 문제점을 극복하고 성능을 개선하고자 하는 것이 상기 FDRT방식이다. 채널 부호화 구조를 사용하는 시스템의 다중접속방식 및 다중채널방식에서 채널 부호화 방식(Channel coding scheme)의 데이터 전송효율성을 높이고 시스템의 성능을 개선하기 위한 전송률정합의 일종인 상기 FDRT방식에 관한 연구가 활발히 진행되었다. 이러한 FDRT방식의 원칙은 사용하는 채널부호가 컨볼루션 부호 또는 선형블록부호 또는 컨볼루션 부호를 사용하는 쉼상부호(Concatenated code)라는 전제하에서 출발한 것이다. 특히 최근에 매우 많은 관심을 모으고 있는 3GPP(3rd Generation Project Partnership 2) IS - 2000의 무선 인터페이스(Air interface)에서 보면 시스템의 다중접속방식 및 다중채널방식에서 채널 부호화 방식의 데이터 전송효율성을 높이고 시스템의 성능을 개선하기 위하여 전송률정합의 일종인 FDRT방식이 표준사항으로 잠정적으로 결정된 상황이며 이에 대한 구현이 진행되고 있는 추세이다.

도 3은 FDRT방식에 따른 전송 장치의 구성을 보여주는 도면이다.

상기 도 3을 설명하기에 앞서서 우선 본문에서 사용하는 용어의 의미가 혼동되는 것을 피하기 위하여 하기의 < 표 1>에 각각 용어의 정의를 하였다. 즉, 도 3에서 $c[n]$, $d[n]$, $f[n]$, $r[n]$ 의 각각은 하기의 < 표 1>에 나타난 바와 같은 데이터 심볼들을 나타낸다. 여기서 심볼이라 함은 하나의 비트로 표기되며 1 또는 0의 값을 가진다. 통상 심볼은 1비트 이상으로 구성되는 경우가 보편적이거나 여기서는 광역의 의미로 1비트로 표시되는 모든 데이터 비트를 심볼로 지칭한다.

[표 1]

용어	정의
$c[n]$	Coded symbols (0..L - 1) from Channel encoder
$r[n]$	Repeated coded symbols (0..LM - 1) by repetition
$f[n]$	Punctured coded symbols (0..N - 1) by FDRT
$d[n]$	Interleaved coded symbols (0..N - 1) by Channel interleaver

참고적으로, FDRT에서는 항상 $L \leq N$ 이므로 항상 Repetition이 되는 것을 의미한다. 왜냐하면, FDRT는 전송하고자 하는 입력데이터 레이트가 IS - 2000 채널 인터리버 크기(Channel interleaver size)와 일치하지 않는 경우 이를 보정하기 위해서 고안된 것이다. 따라서 L 이 $N_{min} < L \leq N$ 인 경우에 효율적인 전송방식을 고려하는 것이므로 항상 $L \leq N$ 이 된다. 따라서 FDRT내에는 천공(puncturing)이 있으나 이는 반복(repetition) 이후에 인터리버 크기(interleaver size) $N = LM - p$ 를 맞추기 위해서 사용되는 것으로 기본적으로 전송되는 전송데이터 심볼(Transmitted symbol)의 수는 부호화된 부호어 심볼(Coded symbol)의 수 L 보다 크다.

상기 도 3을 참조하면, 우선 전송하고자 하는 부호어 심볼수가 채널 인터리버 크기보다 작은 경우 반복기 110은 부호어 심볼을 M번 반복(repetition)한다. IS - 2000의 경우 채널 인터리버(Channel interleaver) 크기는 SF(Spreading Factor)에 따라서 2의 배수로 증가/감소하므로, M은 최소 2가 된다. 수신기에서는 심볼에너지를 증가시키기 위하여 심볼 결합(Symbol combining)을 채널 복호기(Channel decoder) 전단에서 수행한다. 상기 반복기 110에 의하여 반복된 부호어 심볼의 수가 N보다 크므로, 채널 인터리버 100의 크기 N에 맞추기 위해서 천공기 120은 천공(Puncturing)을 수행한다.

도 4는 도 3에 도시된 바와 같이 반복기 110 및 천공기 120에 의해 재구성된 부호어 심볼 프레임의 구조를 도시하는 도면이다.

상기 도 4의 (A)는 하나의 프레임내의 L개의 부호어 심볼을 나타내며, (B)는 상기 반복기 110에 의해 M번 반복된 부호어 심볼, 즉 LM개의 부호어 심볼을 나타낸다. (C)에서 상기 LM개의 부호어 심볼들중 N개의 부호어 심볼들은 상기 채널 인터리버 100에 의해 인터리빙될 심볼들을 나타내고, LM - N개의 부호어 심볼들은 천공될 심볼들임을 나타낸다. 이때 LM - N개의 심볼들은 프레임내에서 균일하게 분포되어 천공되도록 되어 있으며, D거리마다 위치하는 심볼들이 천공된다. (D)는 상기 (C)에 도시된 심볼들중 천공될 심볼들을 천공한 결과에 따른 부호어 심볼들로서 이러한 부호어 심볼들은 채널 인터리버 100으로 인가되어 채널 인터리빙되게 된다.

상기 도 4를 참조하면, 상기 재구성된 부호어 심볼 프레임을 도 2에 도시된 Non FDRT방식의 부호어 심볼 프레임과 비교하면 큰 차이점을 발견할 수 있다. 즉, FDRT방식에서는 프레임 내에 널 데이터가 하나도 없으며 모든 심볼이 부호어 심볼로 처리된다는 점이다. 결국 수신기 입장 측면에서는 Non FDRT방식과는 달리 FDRT방식을 사용하면 동일한 전송 출력전력(Tx power)에서 실제 수신되는 부호어 심볼 에너지(coded symbol energy)가 증가되는 효과를 얻을 수 있다. 상기 부호어 심볼 에너지는 심볼 결합(Symbol combining)된 이후의 부호어 심볼의 에너지를 의미한다. 이러한 효과는 곧 동일한 QoS를 보장하기 위한 기지국의 전송 출력전력(Tx power)을 감소시킬 수 있다는 의미이고, 최종적으로 채널 용량(Channel capacity)을 증가시킬 수 있는 것을 의미한다.

상기 도 4에서 표시된 블록은 천공되는 심볼을 의미하고, D는 천공거리(puncturing distance)를 나타낸다. 상기 천공 거리 D는 LM 심볼들(symbols)로부터 N 심볼(symbol)을 만들기 위해서 수행하는 천공 방식을 결정하는 파라미터이다. 이러한 L, M, N, P, D 등의 관계를 규정하는 FDRT 알고리즘이 되며 이러한 결정방식이 기존의 IS - 2000 FDRT에 제시되어있다.

IS - 2000 사양에서 제시하는 FDRT 알고리즘을 기술하면 하기와 같다. 하기에서는 편의상 원문에서 발췌된 내용, 즉 원래의 영문 용어를 그대로 사용하여 FDRT 알고리즘을 설명하기로 한다. IS - 2000 사양에서 제시하는 FDRT 알고리즘은 하기의 < 표 2> 와 같다.

[표 2]

If variable - rate Reverse Supplemental Channel operation, flexible data rates, or both are supported, puncturing after symbol repetition is calculated as described here. However, the puncturing in 3.1.3.1.6.1 and 3.1.3.1.6.2 is used for the frame formats listed in Table 3.1.3.10.2 - 1 for the Forward Dedicated Control Channel, Table 3.1.3.11.2 - 1 for the Forward Fundamental Channel, or Tables 3.1.3.12.2 - 1, 3.1.3.12.2 - 2, or 3.1.3.12.2 - 3 for the Forward Supplemental Channel. The number of repeated symbols punctured per frame puncturing is defined by $P = LM - N$ where L = Number of specified encoded symbols per frame at encoder output N = Desired channel interleaver size ($N \geq L$) $M = \lceil NL \rceil$ is the symbol repetition factor for flexible data rate If P is equal to 0, then puncturing is not required. If puncturing is necessary, every D th repeated symbol is deleted until the required number of punctured symbols per frame, P , is achieved. That is, if the unpunctured symbols are numbered from 1 to LM , then symbols numbered $D, 2D, 3D, \dots$ are deleted. $D = \lceil LM/P \rceil$ for $P > 0$; otherwise, puncturing is not required.

상기 < 표 2 > 에 나타난 알고리즘에서 보듯이 주어진 파라미터 L, N 으로부터 최종적으로 D 를 구하고 이 D 값을 사용하여 첫 번째 부호어 심볼로부터 차례로 매 D 번째 부호어 심볼들을 천공하여 최종적으로 $LM - N$ 인 P 개의 부호어 심볼들을 천공하게 된다. 그러나 이러한 FDRT 방식은 컨볼루션 부호 (Convolutional code) 의 특성 상 아래와 같은 조건들이 고려되지 않았으므로 성능상에 문제를 일으킬 수 있다.

일반적으로 채널부호화 방식으로 컨볼루션 부호와 단일복호기가 사용되는 선형블록부호 등이 주로 사용되는 경우 이런 채널부호화 구조를 사용하는 시스템의 다중접속방식 및 다중채널방식에서 채널 부호화 방식의 데이터 전송효율성을 높이고 시스템의 성능을 개선하기 위한 가변 데이터 전송율 방식에서 천공을 사용하는 경우 아래의 조건들이 충분히 고려되고 반영되어야 한다.

(조건 1) 입력심볼 시퀀스를 일정한 주기를 갖는 천공 패턴 (puncturing pattern) 으로 천공한다.

(조건 2) 입력심볼의 천공비트 수를 가급적 최소화 한다.

(조건 3) 부호기에서 출력되는 부호어 심볼을 균일한 천공패턴을 사용하여 천공한다.

상기한 조건들은 컨볼루션 부호는 부호기에서 출력되는 부호어 심볼의 에러 감도 (error sensitivity) 가 하나의 프레임 (부호어) 내의 모든 심볼에 대해서 거의 유사하다는 가정에서 출발하였으며, 실제로 위의 방식을 천공방식의 주요 제한요소로 사용하여 긍정적인 결과를 구할 수 있다. 그러나 위에서 제시한 IS - 2000 의 FDRT 방식은 대부분의 경우 위의 조건을 만족하지 못한다.

도 5는 상기와 같은 조건들을 만족하지 못함에 따라 발생하는 문제의 한가지 예를 설명하기 위한 도면이다. 상기 도 5에서 사용된 조건은 하기의 < 표 3 > 과 같다.

[표 3]

IS - 2000 RC3 (Code rate $R=1/4$) Maximum Assigned Data Rate = 19.2kbps $N=1536$ bits Input Data Rate = 15kbps Code d symbols per frame (L) = 1200bits $M = \lceil NL \rceil = \lceil 1536/1200 \rceil = 2$ $P = 864$ bits ($LM - N = 2400 - 1536$) $D = \lceil LM/P \rceil = \lceil 2400/864 \rceil = 2.778$

상기 도 5를 참조하면, 실제로 천공이 부호어 심볼 프레임 (coded symbol frame) 의 앞부분인 1728비트에서만 이루어지고 프레임이 뒷부분인 672비트 구간에서는 전혀 이루어지지 않는 것을 알 수 있다. 참고로 상기 도 5에서 진한색으로 표시된 부분이 천공된 심볼을 의미한다. 또한 점선블록으로 표시된 672개의 심볼은 모두 2번씩 반복되어 전송되고 앞부분의 1728개의 심볼은 2번씩 반복된 것 중 하나만이 선택적으로 전송되며 이를 모두 포함하면 $N=1536$ 심볼이 형

성된다. 이러한 $N=1536$ 의 프레임 내부의 심볼 구조를 볼 때 이것은 위에서 언급한 (조건 3)에 위배되는 것이다. 따라서 이러한 FDRT방식은 불균일한 천공에 의해서 성능의 열화가 발생할 수 있음을 의미한다.

도 6은 종래기술에 따른 FDRT 방식에서 수신기의 종단에서 제공하는 심볼 에너지의 분포 및 단위 프레임당 심볼 수를 도시하는 도면이다.

상기 도 6을 참조하면, FDRT 방식에 따라 전송된 심볼들은 채널 수신기 200에 수신된 후 이레이저 삽입 및 심볼 결합기(Erasure Insertion & Symbol Combining) 210으로 인가된다. 상기 심볼 결합기 210에서 인가되는 심볼들에 대해 심볼 결합(Symbol Combining)하는 경우 각각의 심볼들이 가지는 상대적인 심볼 에너지(Symbol energy: E_s)의 분포가 도시되어 있다. 도면에서 알 수 있듯이 반복되지 않은 864개의 심볼들의 심볼 에너지 E_s 를 1.0으로 정규화하였을 때, 상대적으로 반복된 후미의 672개의 수신 심볼들은 $M=2$ 로 심볼 결합되어 E_s 가 2.0이 된다. 따라서 후미의 심볼들은 동일한 채널환경에서 평균 +3dB의 E_s/N_0 의 이득을 가진다. 결국 이렇게 불균일하게 분포한 1200심볼로부터 $R=1/4$ 채널 복호기(Channel decoder) 220은 복호를 수행하여 300비트의 정보 심볼(information symbol)을 출력한다. 뒤에서 제시하겠지만 실제로 성능 모의실험결과 상당히 큰 성능의 열화가 발생하는 것을 알 수 있다. 따라서 성능을 개선하기 위해서는 이러한 문제점을 해결하는 것이 필요하다.

상기와 같은 불균일한 천공이 발생하는 원인은 바로 천공패턴(Puncturing pattern)을 결정하는 D값에 있다. 즉, 기존의 IS-2000의 FDRT 알고리즘에서 보면 D를 결정할 때 LM/P 가 정수가 아닌 경우에 이 보다 작은 최소의 정수인 $\lfloor LM/P \rfloor$ 을 D로 결정하였다. 따라서 이러한 경우 실제 천공은 $P \times D$ 개 만큼만이 발생하며 나머지 $P \times (LM/P - D)$ 의 구간에서는 천공이 발생하지 않는다. 예를 들어 앞의 예제의 경우 $LM/P=2.778$ 이므로, $D=2$, $LM/P - D=0.778$ 이다. 따라서 $P \times D = 864 \times 2 = 1728$ 에서는 천공이 발생하고 $P \times (LM/P - D) = 864 \times 0.778 = 672$ 의 구간에서는 천공이 발생하지 않는다. 결론적으로 D를 결정하는 과정에서 $(LM/P - D)$ 의 차이로 인해 불균일한 천공이 발생하는 것이다.

전술한 바와 같은 종래기술에 따른 FDRT 방식의 문제점을 정리해보면 다음과 같다.

첫째, 컨볼루션 부호 또는 선형블록부호에 사용되는 FDRT방식은 부호기에서 출력되는 부호어 심볼의 에러 감도가 하나의 프레임(부호어)내의 모든 심볼에 대해서 거의 유사하다는 특징으로부터 가급적 균일한 천공(Uniform puncturing)방식을 요구한다. 그러나 현재 IS-2000의 FDRT방식의 경우에는 이와 같은 가정이 성립되지 않으므로 기존의 FDRT방식을 다르게 적용할 필요가 있다.

둘째, 기존의 IS-2000의 FDRT방식의 경우에는 심볼 반복의 관점에서 기본적으로 FDRT를 반복 방식(Repetition Scheme)으로 간주하여 천공패턴(puncturing pattern)에 크게 영향이 없는 것으로 간주하였으나 이는 천공과 동일한 개념에서 해석되어야 한다. 즉, 반복의 경우에도 최적의 성능의 FDRT방식을 위해서는 부호기에서 출력되는 부호어 심볼의 에러 감도가 하나의 프레임(부호어) 내의 모든 심볼에 대해서 거의 유사하다는 특징으로부터 가급적 균일한 반복(Uniform repetition)방식을 사용해야 한다. 그러나 현재의 IS-2000의 FDRT방식의 경우에는 이와 같은 가정이 성립되지 않으므로 기존의 FDRT방식을 다르게 적용할 필요가 있다.

발명이 이루고자 하는 기술적 과제

따라서 본 발명의 목적은 데이터 통신시스템에서 컨볼루션 부호 또는 선형블록부호를 사용하거나 모두를 사용하는 경우 최적의 성능을 보일 수 있는 가변데이터 전송을 정합 방법 및 장치를 제공함에 있다.

본 발명의 다른 목적은 컨볼루션 부호 또는 선형블록부호를 사용하는 데이터 통신시스템에서 간단한 구조, 그리고 설정 초기값을 조절함으로써 전송율에 따라서 유연하게 동작하는 가변데이터 전송을 정합 방법 및 장치를 제공함에 있다.

이러한 목적들을 달성하기 위한 본 발명에 따르면, 하나의 프레임내에 포함되는 L 개의 코드심볼들이 M 배로 반복됨에 따라 생성되는 LM 개의 심볼들중에서 인터리버 크기에 일치하도록 P 개의 심볼들을 천공될 심볼로 결정하는 방법, 각 행들이 L 개의 코드심볼들로 구성되는 M 개의 행들중에서 1개의 행을 천공을 행할 행으로 선택하는 과정과, 상기 선택된 행에 포함되는 L 개의 코드심볼들중에서 균일하게 분포하는 P 개의 심볼들을 천공 심볼들로 결정하는 과정으로 이루어진다.

본 발명의 제1견지(Aspect)에 따르면, 데이터 통신시스템에서 인터리버 크기와 다른 코드심볼들을 전송하는 가변데이터 전송 방법은, 미리 설정된 프레임내의 상기 코드심볼들을 적어도 상기 인터리버 크기보다 크도록 반복하는 과정과, 상기 반복된 코드심볼들중에서 상기 프레임내에서 균일하게 분포하는 코드심볼들을 천공하도록 하는 천공패턴을 결정하는 과정과, 상기 반복된 코드심볼들을 상기 천공패턴에 따라 천공하는 과정과, 상기 천공된 코드심볼들을 채널 인터리빙하여 전송을 위해 출력하는 과정을 포함한다. 상기 천공패턴은, 상기 프레임내의 코드심볼들중 D_1 및 D_2 위치에 해당하는 심볼들을 천공하기 위한 심볼로 결정한다. 여기서, $D_1 = \lfloor LP \rfloor$ 이고, $D_2 = \lfloor P_1/P_2 \rfloor D_1$ 이고, L 은 부호화기 출력 코드심볼의 수이고, $M = \lfloor N/L \rfloor$ 은 반복 배수이고, $P = LM - \lambda$ 는 천공될 심볼의 수이고, $P_1 = \lfloor LD_1 \rfloor$ 는 첫 번째 천공 주기(D_1)로 천공될 심볼의 수이고, $P_2 = P - P_1$ 은 두 번째 천공 주기(D_2)로 천공될 심볼의 수이다.

본 발명의 제2견지에 따르면, 데이터 통신시스템에서 인터리버 크기와 다른 코드심볼들을 전송하는 가변데이터 전송 방법은, 미리 설정된 프레임내의 상기 코드심볼들을 적어도 상기 인터리버 크기보다 크도록 반복하는 과정과, 상기 반복된 코드심볼들중에서 천공하지 않고 남겨둘 코드심볼들을 나타내는 비천공패턴을 결정하는 과정과, 상기 반복된 코드심볼들중에서 상기 비천공패턴에 해당하지 않는 코드심볼들을 천공하는 과정과, 상기 천공된 코드심볼들을 채널 인터리빙하여 전송을 위해 출력하는 과정을 포함한다. 상기 비천공패턴은, 상기 프레임내의 코드심볼들중 D_1 및 D_2 위치에 해당하는 심볼들을 천공하지 않을 심볼로 결정한다. 여기서, $D_1 = \lfloor LA \rfloor$ 이고, $D_2 = \lfloor A_1/A_2 \rfloor D_1$ 이고, L : 부호화기 출력 코드심볼의 수이고, $M = \lfloor N/L \rfloor$ 은 반복 배수이고, $P = LM - \lambda$ 는 천공될 심볼의 수이고, $A = L - P$ 는 천공되지 않을 심볼의 수이고, $A_1 = \lfloor LD_1 \rfloor$ 은 첫 번째 비천공 주기(D_1)에 의해 남겨질 심볼의 수이고, $A_2 = A - A_1$ 은 두 번째 비천공 주기(D_2)에 의해 남겨질 심볼의 수이다.

발명의 구성 및 작용

이하 본 발명의 바람직한 실시예의 상세한 설명이 첨부된 도면들을 참조하여 설명될 것이다. 도면들 중 참조번호들 및 동일한 구성요소들에 대해서는 비록 다른 도면상에 표시되더라도 가능한 한 동일한 참조번호들 및 부호들로 나타내고 있음에 유의해야 한다. 하기에서 본 발명을 설명함에 있어, 관련된 공지 기능 또는 구성에 대한 구체적인 설명이 본 발명의 요지를 불필요하게 흐릴 수 있다고 판단되는 경우에는 그 상세한 설명을 생략할 것이다.

하기에서 설명될 본 발명은 종래 기술에 따른 가변데이터 전송(FDRT)방식이 지닌 문제점을 해결하기 위해서 균일한 천공(puncturing) 또는 반복(repetition)이 이루어지도록 하는 FDRT방식에 관한 것임을 밝혀두는 바이다. 이는 곧 균일한 천공패턴(Uniform puncturing pattern) 또는 균일한 반복패턴(Uniform repetition pattern)을 필요로 하는 것을 의미한다. 따라서 하기에서 설명될 본 발명은 FDRT를 위한 새로운 천공패턴 생성방식을 제안한다.

우선 본 발명자는 앞서서 설명된 종래 기술에 따른 FDRT 방식에서 균일 천공 또는 균일 반복이 이루어지도록 하는데 있어서 가장 큰 문제가 된 것이 D의 결정에 있음을 인식한다. 다시 부연 설명하면, 불균일한 천공 또는 반복이 발생하는 원인은 바로 천공패턴 또는 반복패턴을 결정하는 D값에 있다. 즉, 종래 기술에 따른 IS-2000의 FDRT 알고리즘에서 보면 D를 결정할 때 LM/P 가 정수가 아닌 경우에 이 보다 작은 최소의 정수인 $\lfloor LM/P \rfloor$ 을 D로 결정하였다. 따라서 이러한 경우 실제 천공은 $P \times D$ 개 만큼만이 발생하며 나머지 $P \times (LM/P - D)$ 의 구간에서는 천공이 발생하지 않는다. 예를 들어, 앞의 예제의 경우 $LM/P=2.778$ 이므로, $D=2$, $LM/P - D=0.778$ 이다. 따라서 $P \times D = 864 \times 2 = 1728$ 에서는 천공이 발생하고, $P \times (LM/P - D) = 864 \times 0.778=672$ 의 구간에서는 천공이 발생하지 않는다. 결론적으로 D를 결정하는 과정에서 $(LM/P - D)$ 의 차이로 인해 불균일한 천공이 발생하는 것이다. 이러한 문제점을 해결하기 위해서 다음과 같은 기본 조건들을 제시하고 이에 따라 아래의 알고리즘을 제안한다.

(FDRT 조건1) L과 N으로부터 결정되는 $P \times D$ 는 $P \times D \geq LM$ 을 만족해야 한다. 즉, D는 $D \geq LM/P$ 을 만족해야 한다. 단, 여기서 P와 D는 정수를 의미한다.

(FDRT 조건2) 상기 (FDRT 조건1)을 만족하는 D로부터 구한 $\lfloor LM/D \rfloor$ 개의 심볼 위치를 제외한 나머지인 $(P - \lfloor LM/D \rfloor)$ 개의 심볼들을 LM개의 심볼들 전체에서 가급적 균일하게 (즉, 등간격이 되도록) 천공 또는 반복한다. 단, 이때 결정되는 심볼 위치는 상기 (FDRT 조건1)에 의해서 구한 D가 결정하는 위치와 반드시 중복되지 않도록 설정한다.

(FDRT 조건3) D를 결정하는 과정에서 $(LM/P - D)$ 의 차이로 인한 불균일한 반복 또는 천공패턴을 최소화한다.

이하에서는 상기 FDRT 조건들을 고려한 본 발명에 따른 FDRT방식 전송 동작에 대하여 설명한다. 먼저 본 발명의 제1실시예에 따른 FDRT방식의 알고리즘이 적용된 예를 설명하고, 다음에 본 발명의 제1실시예가 일반화된 FDRT방식에 따른 전송에 사용될 수 있음을 설명하기로 한다.

A. 제1 실시 예

A New Flexible Data Rate Transmission Algorithm Type 1

본 발명의 제1실시예에 따른 FDRT방식의 알고리즘이 적용된 예를 설명한다. 이 실시예에서 사용된 조건은 하기의 < 표 4> 와 같으며, 알고리즘은 하기의 < 표 5> 와 같다.

[표 4]

IS-2000 RC3 (Code rate $R=1/4$) Maximum Assigned Data Rate = 19.2kbps N=1536bits Input Data Rate = 15kbps Code d symbols per frame (L) = 1200bits M = $\lfloor N/L \rfloor = \lfloor 1536/1200 \rfloor = 2$ P = 864bits (LM - N=2400 - 1536) D = $\lfloor LM/P \rfloor = \lfloor 2400/864 \rfloor = 2.778$ = 3

[표 5]

D = $\lfloor LM/P \rfloor + 3$ The repeated symbol is deleted if the following condition is satisfied. If $(K \bmod 3 = 2 \text{ or } K \bmod 36 = 0)$ then Puncturing where $K=0,1,2,3, \dots, 2399$

상기 < 표 5> 를 참조하면, 본 발명의 제1실시예에 따른 알고리즘에서 $K \bmod 3$ 은 K를 3으로 나눈 나머지를 의미한다. 위 D를 구하는 과정에서 (FDRT 조건1)을 사용하였으며, 36이라는 변수를 구하는 과정에서 (FDRT 조건2)를 사용하였다.

도 7은 본 발명에 의해 제안된 천공패턴에 따라 부호어 심볼 프레임의 천공하는 예를 도시하는 도면이다. 이 예는 상기 < 표 4>에 나타난 조건 및 < 표 5>에 나타난 알고리즘에 따른 것이다.

상기 도 7을 참조하면, 실제로 천공이 부호어 심볼 프레임(coded symbol frame)의 전 구간에서 거의 균일하게 이루어지는 것을 볼 수 있다. 상기 도 7에서 진한색으로 표시된 부분이 천공된 심볼을 의미한다. 또한 2번씩 반복되어 전송되는 심볼과 2번씩 반복된 것 중 하나만이 선택적으로 전송되는 심볼이 균일하게 분포하는 것을 볼 수 있다. 따라서 이러한 $N=1536$ 의 프레임 내부의 심볼구조를 볼 때 이것은 앞서 언급한 (FDRT 조건3)에 부합하는 구조를 가진다. 따라서 이러한 FDRT방식은 균일한 천공에 의해서 성능의 열화가 발생하지 않으며 최적의 성능에 근접하는 성능을 보일 수 있다는 것을 의미한다.

도 8은 본 발명의 일 실시 예에 따른 FDRT 방식에서 수신기의 종단에서 제공하는 심볼 에너지의 분포 및 단위 프레임 당 심볼 수를 도시하는 도면이다.

상기 도 8을 참조하면, 본 발명의 FDRT방식으로 전송된 심볼들을 채널 200을 통해 수신되고 이레이저 삽입 및 심볼 결합기 210으로 인가된다. 상기 심볼 결합기 210에서 심볼을 결합하는 경우 (A)에 도시된 바와 같이 1200개의 심볼들이 출력되며, 이때 출력되는 심볼들은 (B)에 도시된 바와 같이 각각의 심볼들이 가지는 상대적인 심볼에너지(Symbol energy : E_s)의 분포를 갖는다. 그림에서 보듯이 반복되지 않는 864개의 심볼들의 E_s 를 1.0으로 정규화하였을 때, 상대적으로 반복된 672개의 수신 심볼들은 $M=2$ 로 심볼 결합되어 E_s 가 2.0이 되며 이러한 심볼들이 전 구간에 균일하게 분포하는 것을 보여주고 있다. 이러한 균일한 분포는 채널 복호기(Channel decoder)(예: Viterbi decoder) 220의 성능을 개선시킨다.

Generalized Flexible Data Rate Transmission Algorithm GFDRTA - I

본 발명의 제1실시예에 따른 FDRT방식을 일반화한 알고리즘을 설명하면 다음과 같다. 우선 본 발명에 따른 FDRT 알고리즘 및 알고리즘에 사용되는 변수들을 정의하면 하기의 < 표 6>과 같다.

[표 6]

$P = LM - N$ 여기서, L = Number of specified encoded symbols per frame at encoder output N = Desired channel interleaver size ($N \geq L$) $M = \lceil N/L \rceil$ is the symbol repetition factor for flexible data rate

상기 < 표 6>에 나타난 알고리즘에서, 만약 P 가 0이면 천공은 수행되지 않는다. 천공이 수행되면 단위 프레임 당 P 개의 비트가 천공될 때까지 M 번씩 반복된 부호어 심볼들을 매 $(D1)$ 번째와 $(D2+1)$ 번째마다 천공한다. 즉, M 번씩 반복된 부호어 심볼들 1에서 LM 까지 순서를 정한 경우 $D1, 2D1, 3D1, \dots$ 의 순서에 해당하는 부호어 심볼들과 $D2+1, 2D2+1, 3D2+1, \dots$ 의 순서에 해당하는 부호어 심볼들을 천공한다. 이때 $D2+1, 2D2+1, 3D2+1, \dots$ 의 순서는 $mD1$ ($m=1,2,3, \dots$)과의 천공위치가 중복되지 않도록 천공위치가 빗겨나가도록 하기 위한 것이다. 따라서 필요한 경우 $mD1$ ($m=1,2,3, \dots$)과의 천공위치가 중복되지 않도록 하기 위한 다른 방안이 고려될 수도 있다. 예를 들어, $D2+1, 2D2+1, 3D2+1, \dots$ 의 순서에 해당하는 부호어 심볼들을 천공하는 대신에 $D21-1, 2D2-1, 3D2-1, \dots$ 의 순서에 해당하는 부호어 심볼들을 천공하는 경우도 고려될 수 있는데, 이 경우도 $mD1$ ($m=1,2,3, \dots$)과의 천공위치가 중복되지 않도록 하기 위한 것이다. 여기서 사용되는 $D1$ 과 $D2$ 는 하기의 < 수학적식 1>로 결정된다.

수학적식 1
 $D1 = \lceil LM/P \rceil$ for $P > 0$; Otherwise, puncturing is not required.

$$P1 = \lfloor LM/D1 \rfloor$$

$$P2 = P - P1$$

$D2 = sD1$ for $P2 > 0$; Otherwise, puncturing is not required.

상기 < 수학식 1> 에서 s는 아래의 < 수학식 2> 를 만족하는 범위의 정수중에서 최대의 정수를 의미한다.

$$\text{수학식 2} \\ (sD1) \leq \lfloor LMP2 \rfloor ,$$

$$s \leq (\lfloor LMP2 \rfloor)/D1 = (\lfloor LM/D1 \rfloor)/P2 = \lfloor P1/P2 \rfloor$$

도 9는 상기 < 표 6> 에 나타낸 바와 같은 본 발명의 제1실시예에 따른 FDRT 전송 동작의 처리 흐름을 보여주는 도면이다.

상기 도 9를 참조하면, 401단계에서는 FDRT에 필요한 최초의 파라미터들(N,L,M,P)을 초기화한다. L과 N은 주어진 데이터 레이트에 따라서 결정되는 것이며, M, P는 상기 식에서 구해지는 값이다. 402단계에서는 알고리즘에 주어진 식에 따라서 파라미터 D1과 P1을 계산한다. 403단계에서는 알고리즘에 주어진 식에 따라서 파라미터 D2와 P2를 계산한다. 상기 단계에서 파라미터들이 모두 결정되면, 404단계 내지 408단계를 수행하여 k는 1로부터 LM까지 순차적으로 카운팅을 시작한다. 매 카운팅마다 조건문은 405단계에서 k가 (D1의 배수) 또는 (D2의 배수+1)인가를 확인하여 해당하는 경우, 406단계에서 해당 k번째 부호어 심볼을 천공한다. 상기 405단계에서 확인되지 않은 경우에는 상기 406단계를 수행하지 않고, 407단계로 바로 진행하여 k값을 +1 증가시킨다. 그리고 k가 LM까지 모두 카운팅되었는지를 확인하여 아직 수행되어야 하는 k가 있으면 위 동작을 k=LM까지, 즉 408단계에서 k=LM+1인 것으로 확인될 때까지 상기 405단계 내지 407단계의 동작을 반복한다. 이러한 방식에 의해서 거의 균일(Uniform)한 FDRT 천공 패턴(puncturing pattern)이 생성된다.

상기 도 9에서는 k가 (D1의 배수) 또는 (D2의 배수+1)인가를 확인하여 해당하는 경우, 해당 k번째 부호어 심볼들을 천공하는 것으로 설명하였다. 그러나 전술한 바와 같이 k가 (D1의 배수) 또는 (D2의 배수 - 1)에 해당하는 경우 그 부호어 심볼들을 천공할 수도 있다. 이는 D1의 배수에 해당하는 부호어 심볼들과 빗겨나간 다른 위치에서 천공이 이루어지도록 하기 위한 것이다. 즉, (D2의 배수 + 1) 또는 (D2의 배수 - 1)에 해당하는 부호어 심볼들은 D1의 배수의 위치에 대하여 천공되는 부호어 심볼들과 다른 위치에서 천공되는 부호어 심볼들이다. 보다 일반적으로는 (D1의 배수 + K_1) 또는 (D2의 배수 + K_2) (여기서 $K_1 \neq K_2$) 에 해당하는 심볼들을 천공하도록 구현할 수 있다.

도 10 및 도 11은 본 발명의 제1실시예에 따른 가변데이터 전송 장치의 구성을 보여주는 도면이다. 상기 도 10에 도시된 장치는 전술한 FDRT 알고리즘을 하드웨어(H/W)적으로 구현한 예에 해당하고, 도 11은 전술한 FDRT 알고리즘을 소프트웨어(S/W)적으로 구현한 예에 해당한다. 즉 본 발명에 따른 가변데이터 전송 장치는 도 11에 도시된 바와 같이 DSP 혹은 CPU 등의 Module S/W로 구현이 가능하며, 도 10에 도시된 바와 같이 ASIC등의 H/W로도 구현이 가능하다.

상기 도 10을 참조하면, FDRT의 데이터 레이트에 따라서 우선 D1과 D2가 각각의 모듈로 디바이스 레지스터 (Modulo Device Register) 320,330에 설정된다. 그러면 카운터(Counter)로 구현될 수 있는 심볼 인덱스 발생기 310에 의해 발생하는 부호어 심볼 인덱스(Coded symbol index)를 각각의 모듈로 디바이스 (Modulo device) 320,330이 비교해서 나머지가 0인 경우 각각 천공가능신호 (puncturing enable signal) PUNC_EN을 천공기 (puncturing device) 350에 전달한다. 따라서 실제 출력되는 심볼의 수는 LM개중에서 N이 된다.

상기 도 11을 참조하면, 가변데이터 전송 장치를 S/W적으로 구현하는 경우에는 천공기 제어블록 360은 어드레스 발생기 모듈 프로그램 (Address generator module program)을 저장하고 있으며, k가 상기 프로그램에 따른 조건식을 만족하는 경우에는 해당 값을 출력하지 않는다. 그러면 H/W와 동일하게 실제 출력되는 심볼의 수는 LM개 중에서 N이 된다.

이상에서는 본 발명의 제1실시예에 따른 FDRT 방식에 대하여 설명하였다. 하기에서는 본 발명의 제2실시예 및 제3실시예에 따른 FDRT 방식에 대하여 설명하기로 한다. 본 발명의 제2실시예 및 제3실시예는 하기의 (조건 4)를 추가적으로 고려한 것이다. 하기의 (조건 4)를 고려한 본 발명의 제2실시예 및 제3실시예에 따른 FDRT 알고리즘을 본 발명에서는 "보존적 FDRT 알고리즘"이라 칭할 것이다.

(조건 4) 부호화기로부터 출력된 한 심볼이 M 배 반복되었을 때 ($M > 1$), M 개의 반복된 심볼 중 하나 보다 많은 심볼은 천공되지 않는다.

본 발명의 제2실시예 및 제3실시예를 구체적으로 설명함에 있어서 사용되는 정의들은 다음의 < 표 7> 과 같다. 그리고 다음의 < 수학식 3> 에 의해 항상 이 성립한다. P개의 < 수학식 3> 은 프레임 당 천공되는 심볼 수 P가 프레임 당 부호화기 출력 심볼 수 L 보다 항상 작다는 것을 의미한다.

[표 7]

$\lceil X \rceil$ X 보다 크거나 같은 최소 정수 $\lfloor X \rfloor$ X 보다 작거나 같은 최대 정수
--

수학식 3

$$P = LM - N = L \lfloor N/L \rfloor - N < L(N/L + 1) - N = L$$

본 발명의 제2실시예 및 제3실시예를 설명하기에 앞서서 다음과 같은 사항들을 고려한다.

먼저 부호화기로부터 출력된 한 프레임 당 L 개의 코드 심볼들은 각각 M 배로 반복되어 LM 개의 심볼이 된다. LM 개의 심볼 중 P 개의 심볼을 ($P=LM-N$) 천공함으로써 심볼의 개수를 인터리버 크기인 N 개로 ($N=LM-P$) 맞춘다. 이때 천공되는 P 개의 심볼을 선택함에 있어서, L 개의 코드 심볼 각각에 대해 (M-1) 개의 반복된 심볼들은 건드리지 않고, 단 한 개의 반복된 심볼만을 천공 대상으로 고려한다.

도 12는 본 발명의 제2실시예 및 제3실시예에 따른 FDRT 방식을 설명하기 위한 도면이다.

상기 도 12를 참조하여 설명하면, 각 열(column)은 하나의 코드 심볼이 M 배 반복된 M-심볼 그룹을 의미하며, 각 행(row)은 L 개의 코드 심볼을 나타낸다. 이 때 P 개의 심볼을 천공함에 있어서, L 심볼로 구성된 행들 중 M-1 개의 행에 대해서는 심볼 천공이 없으며 하나의 행에 대해서만 심볼 천공이 이루어진다. 상기 하나의 행에서는 L 개의 심볼 중 P 개의 심볼이 천공된다. 따라서 상기 (조건 4)를 만족하는 경우, LM 개의 심볼 중 P 개의 천공할 심볼을 선택하는 문

제를 대신하여, L 개의 심볼 중 P 개의 천공할 심볼을 선택하는 문제를 푸는 것으로 생각할 수 있다. 이상과 같이 반복과 천공이 이루어진 다음 각 심볼들은, 도 12에서 볼 때, 열별로 출력된다. 즉 도 12의 심볼 번호를 참조하면, 1, 1, 2, 2, 3, 4, 4, 5, 5, 6, 7, 7, 8, 8, 9, 10, 10, 11, 11, 12, 13, 13, 14, 14, ..., $L-2$, $L-2$, $L-1$, $L-1$, L 순으로 출력된다.

B. 제2 실시 예

본 발명의 제2실시예에 따른 보존적 FDRT 알고리즘이 설명될 것이다. 우선 본 발명의 제2실시예에 따른 보존적 FDR T 알고리즘 및 알고리즘에 사용되는 변수들을 정의하면 하기의 < 표 8> 과 같다.

< 보존적 FDRT 알고리즘 1 >

L : 부호화기 출력 심볼수

$M = \lceil N/L \rceil$: 반복 배수

$P = LM - N$: 천공될 심볼의 수

$A = L - P$

$D_1 = \lceil L/A \rceil$: 첫 번째 비천공 주기

$A_1 = \lfloor L/D_1 \rfloor$: 첫 번째 비천공 주기에 의해 남겨질 심볼 수

$A_2 = A - A_1$: 두 번째 비천공 주기에 의해 남겨질 심볼 수

$D_2 = \lfloor A_1/A_2 \rfloor D_1$: 두 번째 비천공 주기

여기서, 상기 결정된 변수를 가지고 LM 개의 심볼을 순차적으로 M 개의 심볼단위로 분할한 각각 (반복된 M 개의 심볼로 구성된 그룹)을,

그룹 $G_1, G_2, G_3, \dots, G_k, \dots, G_L$ 이라고 하면,

$$k \in \{ 0 \times D_1 + K_1, 1 \times D_1 + K_1, 2 \times D_1 + K_1, \dots, (A_1 - 1) \times D_1 + K_1, \\ 0 \times D_2 + K_2, 1 \times D_2 + K_2, 2 \times D_2 + K_2, \dots, (A_2 - 1) \times D_2 + K_2 \}$$

에 해당되는 그룹 G_k 의 심볼은 천공하지 않는다.

다음으로 이를 제외한 그룹의 마지막 심볼을 천공한다.

여기서, K_1 은 0 보다 크고 D_1 보다 작거나 같은 정수(e.g. D_1)이고,

K_2 는 0 보다 크고 D_1 보다 작은 정수 (e.g. $\lfloor D_1/2 \rfloor$)이면서

$K_1 \neq K_2$ 을 만족해야 함.

* 이 방법은 부호화된 L 개의 심볼들을 $M-1$ 배 반복한 후, L 개의 심볼들중 A 개의 심볼을 선택해 추가 반복하는 것과 동등하다.

상기 보존적 FDRT 알고리즘 1에서는 L 개의 심볼 중 P 개의 심볼을 천공한다는 개념을 L 개의 심볼 중 P 개의 심볼을 제외한 A 개의 심볼을 천공하지 않고 남겨둔다는 개념과 동일하게 본다. 따라서 상기 보존적 FDRT 알고리즘 1은 L 개의 심볼들 중에 A 개의 남겨둘 (천공하지 않을) 심볼들은 선택하는 과정으로 이루어진다. 남겨둘 심볼들은 두 가지 주기 D_1, D_2 에 따라 각각 A_1, A_2 개가 선택되며, 상기 두 주기 중 D_2 가 D_1 의 배수라도 선택되는 심볼들이 중복되지 않도록 서로 다른 상수 K_1, K_2 만큼 오프셋 (offset) 되어 선택된다. 상기 보존적 FDRT 알고리즘은 부호화된 L 개의 심볼들을 $M-1$ 배 반복한 후, L 개의 심볼들 중 A 개의 심볼을 선택해 추가 반복하는 것과 동등하다.

이하 도 13, 14, 15에 대한 설명에서는 $K_1=D_1$ 을 가정하였다.

도 13은 상기 < 표 8>에 나타낸 바와 같은 본 발명의 제2실시예에 따른 FDRT 전송 동작의 처리 흐름을 보여주는 도면이다.

상기 도 13을 참조하면, 상기 보존적 FDRT 알고리즘 1은, 단계 411, 412, 413에서 N, L, M, 그리고 P 값으로부터 파라미터들 (D_1, A_1, A_2, D_2)을 계산하고, 단계 415에서 인덱스 k에 대한 조건 $k \bmod (M \times D_1) = 0$ OR $k \bmod (M \times D_2) = M \times K_2$ 을 확인하고, "아니오"이면 단계 416에서 천공을 하고, 단계 417에서 인덱스 k를 $k = k + M$ 로 증가(increment)시키고, 단계 415부터의 동작을 k가 LM보다 클 때까지 반복적으로 수행하는 단계로 구현될 수 있다.

도 14 및 도 15는 본 발명의 제2실시예에 따른 가변데이터 전송 장치의 구성을 보여주는 도면이다. 상기 도 14에 도시된 장치는 전술한 보존적 FDRT 알고리즘 1을 하드웨어(H/W)적으로 구현한 예에 해당하고, 도 15는 전술한 보존적 FDRT 알고리즘 1을 소프트웨어(S/W)적으로 구현한 예에 해당한다. 즉 본 발명에 따른 가변데이터 전송 장치는 도 15에 도시된 바와 같이 DSP 혹은 CPU 등의 Module S/W로 구현이 가능하며, 도 10에 도시된 바와 같이 ASIC등의 H/W로도 구현이 가능하다.

상기 도 14를 참조하면, 보존적 FDRT 알고리즘 1의 경우 모듈로 장치 321, 331을 $\bmod(M \times D_1), \bmod(M \times D_2)$ 로 함으로써 구현될 수 있다. 상기 보존적 FDRT 알고리즘 1은 모듈로 장치 321이 $k \bmod (M \times D_1) = 0$ 의 논리 연산 결과를 출력하도록 하고, 상기 모듈로 장치 331이 $k \bmod (M \times D_2) = M \times K_2$ 의 논리 연산 결과를 출력하도록 하고, 두 출력에 각각 논리 역 "NOT"을 취한 값들이 "AND" 연산기 341에 가해지도록 함으로써 구현할 수 있다.

상기 도 15를 참조하면, 가변데이터 전송 장치를 S/W적으로 구현하는 경우에는 천공기 제어블록 361은 어드레스 발생기 모듈 프로그램(Address generator module program)을 저장하고 있으며, k가 상기 프로그램에 따른 조건식을 만족하는 경우에는 해당 값을 출력하지 않는다. 그러면 H/W와 동일하게 실제 출력되는 심볼의 수는 LM개 중에서 N이 된다. 상기 보존적 FDRT 알고리즘 1의 경우, 천공기 제어 블록 361의 출력 PUNC_EN은 $k \bmod M = 0$ 이 아닌 경우에는 항상 0이 되도록 한다. $k \bmod M = 0$ 인 경우, 상기 천공기 제어 블록 361에서 PUNC_EN이 1이 되는 조건을 $k \bmod M = 0$ 로 함으로써 구현할 수 있다.

C. 제3 실시 예

본 발명의 제2실시예에 따른 보존적 FDRT 알고리즘이 설명될 것이다. 우선 본 발명의 제2실시예에 따른 보존적 FDRT 알고리즘 및 알고리즘에 사용되는 변수들을 정의하면 하기의 < 표 9>와 같다.

< 보존적 FDRT 알고리즘 2 >

L : 부호화기 출력 심볼수

$M = \lceil N/L \rceil$: 반복 배수

$P = LM - N$: 천공될 심볼의 수

$D_1 = \lceil L/P \rceil$: 첫 번째 천공 주기

$P_1 = \lfloor L/D_1 \rfloor$: 첫 번째 천공 주기로 천공될 심볼의 수

$P_2 = P - P_1$: 두 번째 천공 주기로 천공될 심볼의 수

$D_2 = \lfloor P_1/P_2 \rfloor D_1$: 두 번째 천공 주기

여기서, 상기 결정된 변수를 가지고 LM 개의 심볼을 순차적으로 M 개의 심볼단위로 분할한 각각 (반복된 M 개의 심볼로 구성된 그룹)을,

그룹 $G_1, G_2, G_3, \dots, G_k, \dots, G_L$ 이라고 하면,

$$k \in \{ 0 \times D_1 + K_1, 1 \times D_1 + K_1, 2 \times D_1 + K_1, \dots, (P_1 - 1) \times D_1 + K_1, \\ 0 \times D_2 + K_2, 1 \times D_2 + K_2, 2 \times D_2 + K_2, \dots, (P_2 - 1) \times D_2 + K_2 \}$$

에 해당하는 그룹 G_k 의 마지막 심볼을 천공한다.

여기서

K_1 은 0 보다 크고 D_1 보다 작거나 같은 정수 (e.g. D_1)

K_2 는 0 보다 크고 D_1 보다 작은 정수 (e.g. $\lfloor D_1/2 \rfloor$)

이면서 $K_1 \neq K_2$ 을 만족해야 함.

예를 들어, ML 개의 입력 심볼들을 아래와 같이 a_i 로 정의하면,

$$S = \{ a_1, a_2, a_3, \dots, a_i, \dots, a_{LM} \}$$

아래의 심볼들만을 천공한다.

$$\{ a_{MK_1}, a_{M(D_1-K_1)}, a_{M(2D_1-K_1)}, a_{M(3D_1-K_1)}, \dots, a_{M((P_1-1)D_1-K_1)} \\ a_{MK_2}, a_{M(D_2-K_2)}, a_{M(2D_2-K_2)}, a_{M(3D_2-K_2)}, \dots, a_{M((P_2-1)D_2-K_2)} \}$$

상기 보존적 FDRT 알고리즘 2에서는 L 개의 심볼 중 P 개의 심볼을 천공한다. 따라서 상기 보존적 FDRT 알고리즘 2 는 L 개의 심볼들 중에 P 개의 천공할 심볼들은 선택하는 과정으로 이루어진다. 천공할 심볼들은 두 가지 주기 D_1, D_2 에 따라 각각 P_1, P_2 개가 선택되며, 상기 두 주기 중 D_2 가 D_1 의 배수라도 선택되는 심볼들이 중복되지 않도록 서로 다른 상수 K_1, K_2 만큼 오프셋 (offset) 되어 선택된다.

이하 도 16, 17, 18에 대한 설명에서는 $K_1 = D_1$ 을 가정하였다.

도 16은 상기 < 표 9>에 나타낸 바와 같은 본 발명의 제3실시예에 따른 FDRT 전송 동작의 처리 흐름을 보여주는 도면이다.

상기 도 16을 참조하면, 단계들 421, 422, 423에서 N, L, M, 그리고 P 값으로부터 파라미터들 (D_1, P_1, P_2, D_2)을 계산하고, 425단계에서 인덱스 k에 대한 조건 $k \bmod (M \times D_1) = 0$ OR $k \bmod (M \times D_2) = M \times K_2$ 을 확인하고, "예" 이면 426단계에서 천공을 하고, 427단계에서 인덱스 k를 $k = k + M$ 로 증가(increment) 시키고, 425단계부터를 k가 LM 보다 클 때까지 반복하는 단계로 구현될 수 있다.

도 17 및 도 18은 본 발명의 제3실시예에 따른 가변데이터 전송 장치의 구성을 보여주는 도면이다. 상기 도 17에 도시된 장치는 전술한 보존적 FDRT 알고리즘 2를 하드웨어(H/W)적으로 구현한 예에 해당하고, 도 18은 전술한 FDRT 알고리즘 2를 소프트웨어(S/W)적으로 구현한 예에 해당한다. 즉 본 발명에 따른 가변데이터 전송 장치는 도 18에 도시된 바와 같이 DSP 혹은 CPU 등의 Module S/W로 구현이 가능하며, 도 17에 도시된 바와 같이 ASIC등의 H/W로도 구현이 가능하다.

상기 도 17을 참조하면, 상기 보존적 FDRT 알고리즘 2의 경우 모듈로 장치 322, 332를 $\bmod(M \times D_1)$, $\bmod(M \times D_2)$ 로 함으로써 구현될 수 있다. 상기 모듈로 장치 322가 $k \bmod (M \times D_1) = 0$ 의 논리 연산 결과를 출력하도록 하고, 모듈로 장치 332가 $k \bmod (M \times D_2)$ 의 논리 연산 결과를 출력하도록 하고, 두 출력이 "OR" 연산기 340에 가해지도록 함으로써 구현할 수 있다.

상기 도 18을 참조하면, 가변데이터 전송 장치를 S/W적으로 구현하는 경우에는 천공기 제어블록 362는 어드레스 발생기 모듈 프로그램(Address generator module program)을 저장하고 있으며, k가 상기 프로그램에 따른 조건식을 만족하는 경우에는 해당 값을 출력하지 않는다. 그러면 H/W와 동일하게 실제 출력되는 심볼의 수는 LM개 중에서 N이 된다. 상기 보존적 FDRT 알고리즘 2의 경우, 천공기 제어 블록 362의 출력 PUNC_EN은 $k \bmod M = 0$ 이 아닌 경우에는 항상 0이 되도록 한다. 보존적 FDRT 알고리즘 2의 경우, 도 11b에 나타낸 바와 같이, 블록 360에서 PUNC_EN이 1이 되는 조건을 $k \bmod M = 0$ AND $\{ k \bmod (M \times D_1) = 0$ OR $k \bmod (M \times D_2) = M \times K_2 \}$ 로 함으로써 구현할 수 있다.

D. 본 발명과 기존의 가변데이터 전송방식과의 관계 및 최적의 성능 분석

다음은 기존의 가변데이터 전송방식과의 관계 및 최적의 성능을 분석하기로 한다. 여기서의 본 발명의 제1실시예에 따른 가변데이터 전송방식과 기존의 가변데이터 전송방식과의 관계 및 최적의 성능을 분석하기로 한다.

여기서는 컨볼루션 부호에서 천공에 따른 성능의 변화를 이론적으로 분석하고, 천공 레이트(Puncturing rate) 또는 반복 레이트(Repetition rate)에 따른 부호율(Code rate)이 R인 컨볼루션 코드의 성능변화의 평균값을 제시한다. 이 평균성능으로부터 앞서 검토한 종래 기술에 따른 IS - 2000의 FDRT 알고리즘과 본 발명에서 제안한 FDRT알고리즘 사이의 성능 차이와 우리가 얻을 수 성능의 평균값을 예측한다. 물론 이러한 성능의 실제 값은 시뮬레이션에 의해서 정확히 비교되므로 다음에서 설명하는 시뮬레이션 결과 부분에서 아래에 계산된 평균성능 예상치와 비교한다.

우선, 아래와 같이 기호를 정의하기로 하자.

R : 사용하는 Convolutional codes의 code rate(=k/n)

R_{st} : 실제 채널로 전송하는 부호어 심볼의 전송레이트 $\times (R)$. 즉, $NR(\text{bits/sec})$

R_{fdrt} : FDRT를 사용할 때 채널부호기에서 출력되는 부호어 심볼 레이트 $\times (R)$. 즉, $LR(\text{bits/sec})$

그러면 균일한 천공 또는 균일한 반복 패턴을 사용하는 것을 가정하는 경우, 천공 또는 반복에 의해 발생하는 성능의 변화는 하기의 < 수학적식 4> 과 같이 주어진다. 여기서 $R_{fdrt} < R_{st}$ 인 경우에는 FDRT방식에서 심볼 반복이 사용되므로, 성능의 개선 즉, 부호화 이득(Coding gain)측면에서 이득이 있다. 그러나 역으로 $R_{fdrt} > R_{st}$ 인 경우에는 심볼 천공이 Symbol Puncturing이 사용되므로, 성능의 열화 즉, 부호화 이득 측면에서 손실을 가진다. 앞서 이야기했듯이 FDRT는 기본적으로 $N > L$ 이므로 반복이 사용되는 구조이며 따라서 심볼 반복이 사용되므로 성능의 개선 즉, 부호화 이득 측면에서 이득이 있다. 문제는 패턴에 따라서 그 부호화 이득 측면에서 이득이 얼마나 달성되느냐 하는 것이다.

수학적식 4

$$\text{Average Coding Gain} = 10\log (R_{st}/R_{fdrt}) \text{ (dB)}$$

일예로, 앞서 분석한 $R_{st}=19.2\text{kbps}$ 인 경우 각각의 R_{fdrt} 에 따른 부호화 이득(Coding Gain)을 하기의 < 표 10> 에 나타냈다. 따라서 만일 천공 패턴 또는 반복 패턴이 적절하게 결정된다면 FDRT를 사용하면 < 표 10> 에 나타낸 바와 같은 부호화 이득이 보장되어야 한다.

[표 10]

구분	R_{st}	R_{fdrt}	
Average Coding Gain			
CASE1	19.2kbps	17.5kbps	
0.40 (dB)			
CASE2	19.2kbps	15kbps	
1.07 (dB)			
CASE3	19.2kbps	10kps	
2.83 (dB)			

도 19 및 도 20은 본 발명의 제1실시예에서 제안하는 FDRT에 따른 시뮬레이션 결과를 도시하는 도면이다.

상기 도 19는 본 발명이 IS - 2000 RC3(Code Rate $R=1/4$)에 적용된 경우에 얻어지는 시뮬레이션 결과를 도시하는 그래프이다. 이러한 그래프는 다음과 같은 시뮬레이션 환경(Simulation Environment)하에서 얻어진다. 각 경우, 즉 CASE1), CASE2), REFERENCE)에 있어서 시뮬레이션 환경은 하기의 < 표 11> , < 표 12> 및 < 표 13> 과 같다.

[표 11]

<p>CASE 1) · Data Rate 15kbps (Pure info + CRC + Tail bit = 300 bit) · L (Encoded size) = 1200 · M = 2, N (Channel Interleaver Size) = 1536, · P (Num of Puncturing) = $L \times M - N = 864$ · D (Puncturing Depth) = $\lfloor (L \times M)/P \rfloor - 2$ · Puncturing Pattern → 비교대상 : 15 kbps signal (Puncturing symbol 개수 동일). → Puncturing Pattern 15KBER_SEC, 15KFER_SEC: NEW Algorithm Type 1사용 if $(k\%3=2 \ \ k\%36=0)$ ($k=0,1,2, \dots, 2399$) then puncturing 15KBER_IS2000, 15KFER_IS2000: 기존의 puncturing pattern사용</p>

[표 12]

CASE 2) · Data Rate 10kbps (Pure info + CRC + Tail bit = 200 bit) · L (Encoded size) = 800 · M = 2, N (Channel Interleaver Size) = 1536, · P (Num of Puncturing) = $L \times M - N = 64$ · D (Puncturing Depth) = $\lfloor (L \times M) / P \rfloor = 25$ · 기존의 Puncturing pattern 사용

[표 13]

REFERENCE) · Reference Curve : 19.2kbps, No Puncturing, No Repetition.

상기 도 19를 참조하면, RC3시뮬레이션결과에서 보듯이 본 발명에서 제안한 FDRT방식이 기존의 IS - 2000의 FDRT 방식에 비하여 약 0.9dB에서 1.0dB의 Eb/No 이득을 제공하고 있다. 이것은 앞서 < 표 10> 에서 분석한 바와 같이 19.2kbps에 비해서 Average Coding Gain 1.07dB에 거의 근접하는 성능이다. 이것은 Puncturing과 Repetition에서 균일한 분포를 가지도록 패턴을 발생한 결과이며 그 성능 또한 최적의 성능에 거의 근접한 성능을 보이고 있다. 따라서 본 특허에서 제안한 FDRT알고리즘의 (FDRT조건1), (FDRT조건2)가 성능에 매우 중요한 역할을 하며 이를 반영한 New FDRT Algorithm Type 1역시 우수한 성능을 제공할 수 있다. 반면에 기존의 IS - 2000의 FDRT알고리즘을 사용한 경우의 결과는 의외로 약 0.1dB 정도의 부호화 이득(Coding gain)만을 제공하고 있음을 알 수 있다. 이것은 앞서 언급하였듯이 전체 프레임 중 후미에 집중된 비대칭 패턴(Asymmetric pattern) 때문에 발생하는 문제이다. 결론적으로 동일한 채널조건에서 FDRT 패턴(Pattern)에 따라서 약 0.9 - 1.0dB의 성능 차이가 발생한다.

상기 도 20은 본 발명이 RC4 SCH(Code Rate R=1/2)에 적용된 경우에 얻어지는 시뮬레이션 결과를 도시하는 그래프이다. 이러한 그래프는 다음과 같은 시뮬레이션 환경(Simulation Environment)하에서 얻어진다. 각 경우, 즉 CASE1), CASE2), CASE3)에 있어서 시뮬레이션 환경은 하기의 < 표 14> , < 표 15> 및 < 표 16> 과 같다.

[표 14]

CASE 1) · Data Rate 15kbps (Pure info + CRC + Tail bit = 300 bit) · L (Encoded size) = 600 · M = 2, N (Channel Interleaver Size) = 768, · P (Num of Puncturing) = $L \times M - N = 432$ · D (Puncturing Depth) = $\lfloor (L \times M) / P \rfloor = 2$ · Puncturing Pattern → 비교대상 : 15 kbps signal (Puncturing symbol 개수 동일). → Puncturing Pattern 15KBER_SEC, 15KFER_SEC: New FDRT Algorithm Type 1사용 if (k%3=2 || k%36=0) (k=0,1,2, ...,2399) then puncturing 15KBER_IS2000, 15KFER_IS2000 : 기존의 puncturing pattern사용

[표 15]

CASE 2) · Data Rate 17.5kbps (Pure info + CRC + Tail bit = 350 bit) · L (Encoded size) = 700 · M = 2, N (Channel Interleaver Size) = 768, · P (Num of Puncturing) = $L \times M - N = 632$ · D (Puncturing Depth) = $\lfloor (L \times M) / P \rfloor = 2$ · 기존의 Puncturing pattern 사용

[표 16]

CASE 3) · Data Rate 10kbps (Pure info + CRC + Tail bit = 200 bit) · L (Encoded size) = 400 · M = 2, N (Channel Interleaver Size) = 768, · P (Num of Puncturing) = $L \times M - N = 32$ · D (Puncturing Depth) = $\lfloor (L \times M) / P \rfloor = 25$ · 기존의 Puncturing pattern 사용

상기 도 20을 참조하면, RC4 시뮬레이션결과도 상기 도 19에 도시된 바와 같이 동일한 결과가 얻어짐을 알 수 있다. 상기 도 13에서 보듯이 본 발명에서 제안한 FDRT방식이 기존의 IS - 2000의 FDRT방식에 비하여 약 0.8dB에서 0.9 dB의 Eb/No 이득을 제공하고 있다.

다음으로 한가지 중요한 것은 10kbps의 성능이다. 이 경우는 기존의 FDRT알고리즘이 < 표 10>에 제시한 Average Coding Gain 2.83dB에 거의 근접하고 있다. 이러한 결과는 10kbps이 경우 D가 정확하게 정수로 결정되기 때문에 앞서 제시한 D를 결정하는 과정에서 (LM/P - D)의 차이로 인해 불균일한 천공이 발생하지 않기 때문이다. 따라서 이는 앞서 본 발명에서 제안한 D의 결정방식 과정에서 (LM/P - D)의 차이를 모두 고려해야 한다는 전제조건이 바로 성능에 직결됨을 잘 보여주는 예라고 할 수 있다. 이 성능에 따른 시뮬레이션 환경은 하기의 < 표 17>과 같다.

[표 17]

<p>· Data Rate 10kbps (Pure info + CRC + Tail bit = 200 bit) · L (Encoded size) = 800 · M = 2, N (Channel Interleaver Size) = 1536, · P (Num of Puncturing) = LM - N = 64 · D (Puncturing Depth) = $\lfloor (L \times M) / P \rfloor = \lfloor 1600 / 64 \rfloor = \lfloor 25 \rfloor = 25$ · 기존의 Puncturing pattern 사용</p>

한편 본 발명의 상세한 설명에서는 구체적인 실시 예에 관해 설명하였으나, 본 발명의 범위에서 벗어나지 않는 한도 내에서 여러 가지 변형이 가능함은 물론이다. 그러므로 본 발명의 범위는 설명된 실시 예에 국한되어 정해져서는 안되며 후술하는 특허청구의 범위뿐만 아니라 이 특허청구의 범위와 균등한 것들에 의해 정해져야 한다.

발명의 효과

상술한 바와 같이 본 발명은 컨볼루션 부호 또는 선형블록부호를 사용하는 데이터 통신시스템에서 간단한 구조, 그리고 설정 초기값을 조절함으로써 천공 패턴 또는 반복 패턴이 프레임내에서 균일하게 분포되도록 함으로써 성능의 열화 없이 전송율에 따라서 데이터를 유연하게 전송할 수 있다는 이점이 있다.

(57) 청구의 범위

청구항 1.

하나의 프레임내에 포함되는 L개의 코드심볼들이 M배로 반복됨에 따라 생성되는 LM개의 심볼들중에서 인터리버 크기에 일치하도록 P개의 심볼들을 천공될 심볼로 결정하는 방법에 있어서,

각 행들이 L개의 코드심볼들로 구성되는 M개의 행들중에서 1개의 행을 천공을 행할 행으로 선택하는 과정과,

상기 선택된 행에 포함되는 L개의 코드심볼들중에서 균일하게 분포하는 P개의 심볼들을 천공 심볼들로 결정하는 과정을 포함함을 특징으로 하는 상기 방법.

청구항 2.

데이터 통신시스템에서 인터리버 크기와 다른 코드심볼들을 전송하는 가변데이터 전송 방법에 있어서,

미리 설정된 프레임내의 상기 코드심볼들을 적어도 상기 인터리버 크기보다 크도록 반복하는 과정과,

상기 반복된 코드심볼들중에서 상기 프레임내에서 균일하게 분포하는 코드심볼들을 천공하도록 하는 천공패턴을 결정하는 과정과,

상기 반복된 코드심볼들을 상기 천공패턴에 따라 천공하는 과정과,

상기 천공된 코드심볼들을 채널 인터리빙하여 전송을 위해 출력하는 과정을 포함하며,

상기 천공패턴은, 상기 프레임내의 코드심볼들중 D1 및 D2 위치에 해당하는 심볼들을 천공하기 위한 심볼로 결정하는 것을 특징으로 하는 상기 가변데이터 전송 방법.

여기서, $D_1 = \lfloor L/P \rfloor$ 이고, $D_2 = \lfloor P_1/P_2 \rfloor D_1$ 이고, L 은 부호화기 출력 코드심볼의 수이고, $M = \lfloor N/L \rfloor$ 은 반복 배수이고, $P = LM - \lambda$ 는 천공될 심볼의 수이고, $P_1 = \lfloor LD_1 \rfloor$ 는 첫 번째 천공 주기(D1)로 천공될 심볼의 수이고, $P_2 = P - P_1$: 두 번째 천공 주기(D2)로 천공될 심볼의 수이다.

청구항 3.

데이터 통신시스템에서 인터리버 크기와 다른 코드심볼들을 전송하는 가변데이터 전송 장치에 있어서,

미리 설정된 프레임내의 상기 코드심볼들을 적어도 상기 인터리버 크기보다 크도록 반복하는 반복기와,

상기 반복된 코드심볼들중에서 상기 프레임내에서 균일하게 분포하는 코드심볼들을 천공하도록 하는 천공패턴을 결정하는 천공패턴 결정기와,

상기 반복된 코드심볼들을 상기 천공패턴에 따라 천공하는 천공기와,

상기 천공된 코드심볼들을 채널 인터리빙하여 전송을 위해 출력하는 채널 인터리버를 포함하며,

상기 천공패턴은, 상기 프레임내의 코드심볼들중 D1 및 D2 위치에 해당하는 심볼들을 천공하기 위한 심볼로 결정하는 것을 특징으로 하는 상기 가변데이터 전송 장치.

여기서, $D_1 = \lfloor L/P \rfloor$ 이고, $D_2 = \lfloor P_1/P_2 \rfloor D_1$ 이고, L 은 부호화기 출력 코드심볼의 수이고, $M = \lfloor N/L \rfloor$ 은 반복 배수이고, $P = LM - \lambda$ 는 천공될 심볼의 수이고, $P_1 = \lfloor LD_1 \rfloor$ 는 첫 번째 천공 주기(D1)로 천공될 심볼의 수이고, $P_2 = P - P_1$: 두 번째 천공 주기(D2)로 천공될 심볼의 수이다.

청구항 4.

제3항에 있어서, 상기 천공패턴 결정기는,

상기 프레임내의 심볼들을 나타내는 인덱스를 발생하는 심볼 인덱스 발생기와,

상기 발생된 인덱스가 상기 프레임내의 심볼들중 D1 위치의 심볼을 나타내는지 및 D2 위치의 심볼을 나타내는지를 판단하는 판단부를 포함하며,

상기 판단부의 판단결과는 상기 천공패턴으로 결정되어 상기 천공기로 제공되는 것을 특징으로 하는 상기 가변데이터 전송 장치.

청구항 5.

제4항에 있어서, 상기 판단부는,

상기 발생된 인덱스가 상기 프레임내의 심볼들중 D1 위치의 심볼을 나타내는지를 판단하는 제1판단부와,

상기 발생된 인덱스가 상기 프레임내의 심볼들중 D2 위치의 심볼을 나타내는지를 판단하는 제2판단부와,

상기 제1판단부 및 상기 제2판단부의 판단결과를 조합하여 상기 천공기로 제공하는 연산기를 포함함을 특징으로 하는 상기 가변데이터 전송 장치.

청구항 6.

데이터 통신시스템에서 인터리버 크기와 다른 코드심볼들을 전송하는 가변데이터 전송 방법에 있어서,

미리 설정된 프레임내의 상기 코드심볼들을 적어도 상기 인터리버 크기보다 크도록 반복하는 과정과,

상기 반복된 코드심볼들중에서 천공하지 않고 남겨둘 코드심볼들을 나타내는 비천공패턴을 결정하는 과정과,

상기 반복된 코드심볼들중에서 상기 비천공패턴에 해당하지 않는 코드심볼들을 천공하는 과정과,

상기 천공된 코드심볼들을 채널 인터리빙하여 전송을 위해 출력하는 과정을 포함하며,

상기 비천공패턴은, 상기 프레임내의 코드심볼들중 D1 및 D2 위치에 해당하는 심볼들을 천공하지 않을 심볼로 결정하는 것을 특징으로 하는 상기 가변데이터 전송 방법.

여기서, $D_1 = \lfloor L/A \rfloor$ 이고, $D_2 = \lfloor A_1/A_2 \rfloor D$ 이고, L : 부호화기 출력 코드심볼의 수이고, $M = \lfloor N/L \rfloor$ 은 반복 배수이고, $P = LM - \Lambda$ 는: 천공될 심볼의 수이고, $A = L - P$ 는 천공되지 않을 심볼의 수이고, $A_1 = \lfloor L/D_1 \rfloor$ 은 첫 번째 비천공 주기(D1)에 의해 남겨질 심볼의 수이고, $A_2 = A - A_1$ 은 두 번째 비천공 주기(D2)에 의해 남겨질 심볼의 수이다.

청구항 7.

데이터 통신시스템에서 인터리버 크기와 다른 코드심볼들을 전송하는 가변데이터 전송 장치에 있어서,

미리 설정된 프레임내의 상기 코드심볼들을 적어도 상기 인터리버 크기보다 크도록 반복하는 반복기와,

상기 반복된 코드심볼들중에서 천공하지 않고 남겨둘 코드심볼들을 나타내는 비천공패턴을 결정하는 비천공패턴 결정기와,

상기 반복된 코드심볼들중에서 상기 비천공패턴에 해당하지 않는 코드심볼들을 천공하는 천공기와,

상기 천공된 코드심볼들을 채널 인터리빙하여 전송을 위해 출력하는 채널 인터리버를 포함하며,

상기 비천공패턴은, 상기 프레임내의 코드심볼들중 D1 및 D2 위치에 해당하는 심볼들을 천공하지 않고 남겨둘 심볼들로 결정하는 것을 특징으로 하는 상기 가변데이터 전송 장치.

여기서, $D_1 = \lfloor L/A \rfloor$ 이고, $D_2 = \lfloor A_1/A_2 \rfloor D$ 이고, L : 부호화기 출력 코드심볼의 수이고, $M = \lfloor N/L \rfloor$ 은 반복 배수이고, $P = LM - \Lambda$ 는: 천공될 심볼의 수이고, $A = L - P$ 는 천공되지 않을 심볼의 수이고, $A_1 = \lfloor L/D_1 \rfloor$ 은 첫 번째 비천공 주기(D1)에 의해 남겨질 심볼의 수이고, $A_2 = A - A_1$ 은 두 번째 비천공 주기(D2)에 의해 남겨질 심볼의 수이다.

청구항 8.

제7항에 있어서, 상기 비천공패턴 결정기는,

상기 프레임내의 심볼들을 나타내는 인덱스를 발생하는 심볼 인덱스 발생기와,

상기 발생된 인덱스가 상기 프레임내의 심볼들중 D1 위치의 심볼을 나타내는지 및 D2 위치의 심볼을 나타내는지를 판단하는 판단부를 포함하며,

상기 판단부의 판단결과는 상기 비천공패턴으로 결정되어 상기 천공기로 제공되는 것을 특징으로 하는 상기 가변데이터 전송 장치.

청구항 9.

제8항에 있어서, 상기 판단부는,

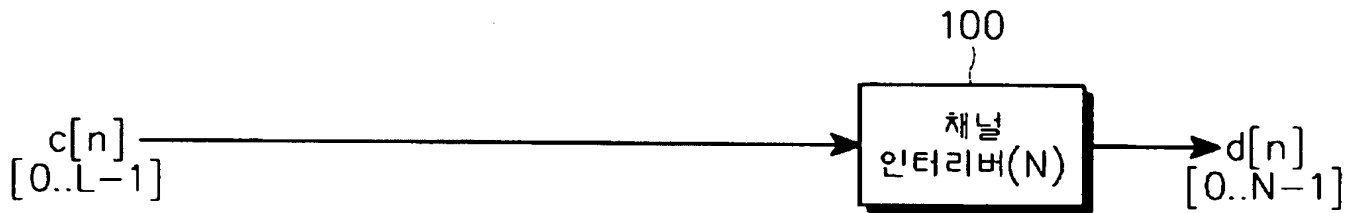
상기 발생된 인덱스가 상기 프레임내의 심볼들중 D1 위치의 심볼을 나타내는지를 판단하는 제1판단부와,

상기 발생된 인덱스가 상기 프레임내의 심볼들중 D2 위치의 심볼을 나타내는지를 판단하는 제2판단부와,

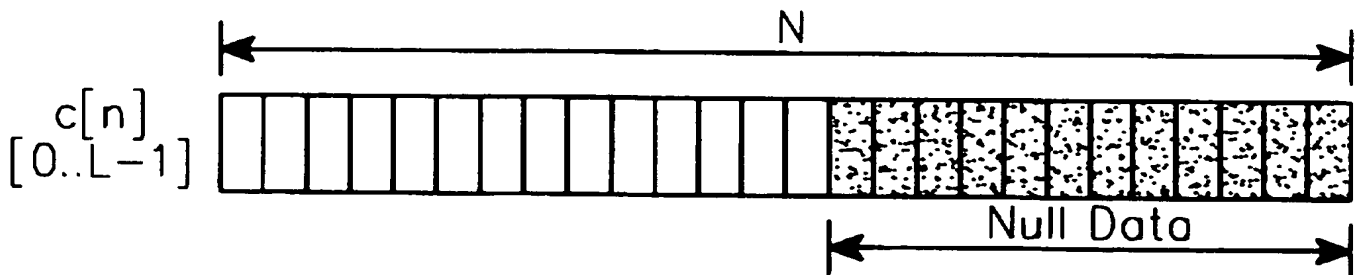
상기 제1판단부 및 상기 제2판단부의 판단결과를 조합하여 상기 천공기로 제공하는 연산기를 포함함을 특징으로 하는 상기 가변데이터 전송 장치.

도면

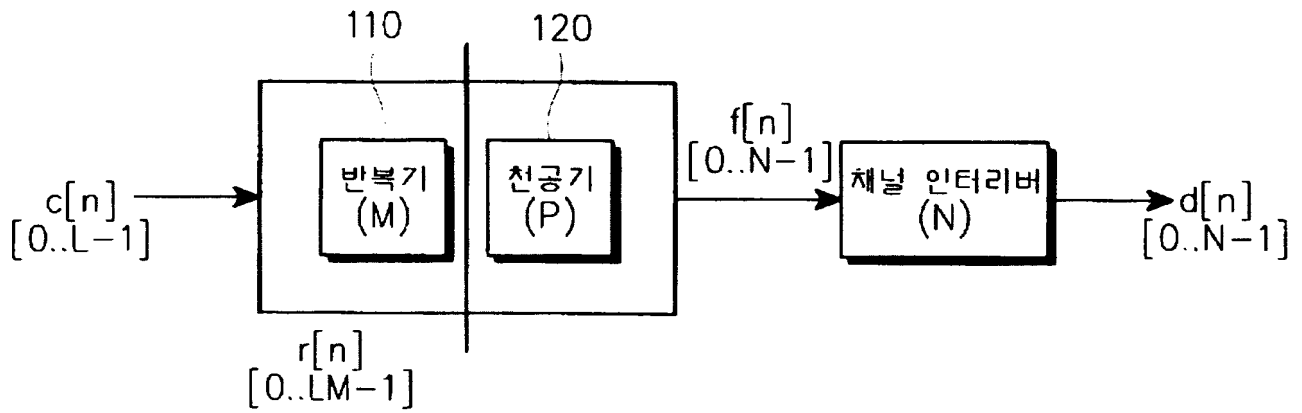
도면 1



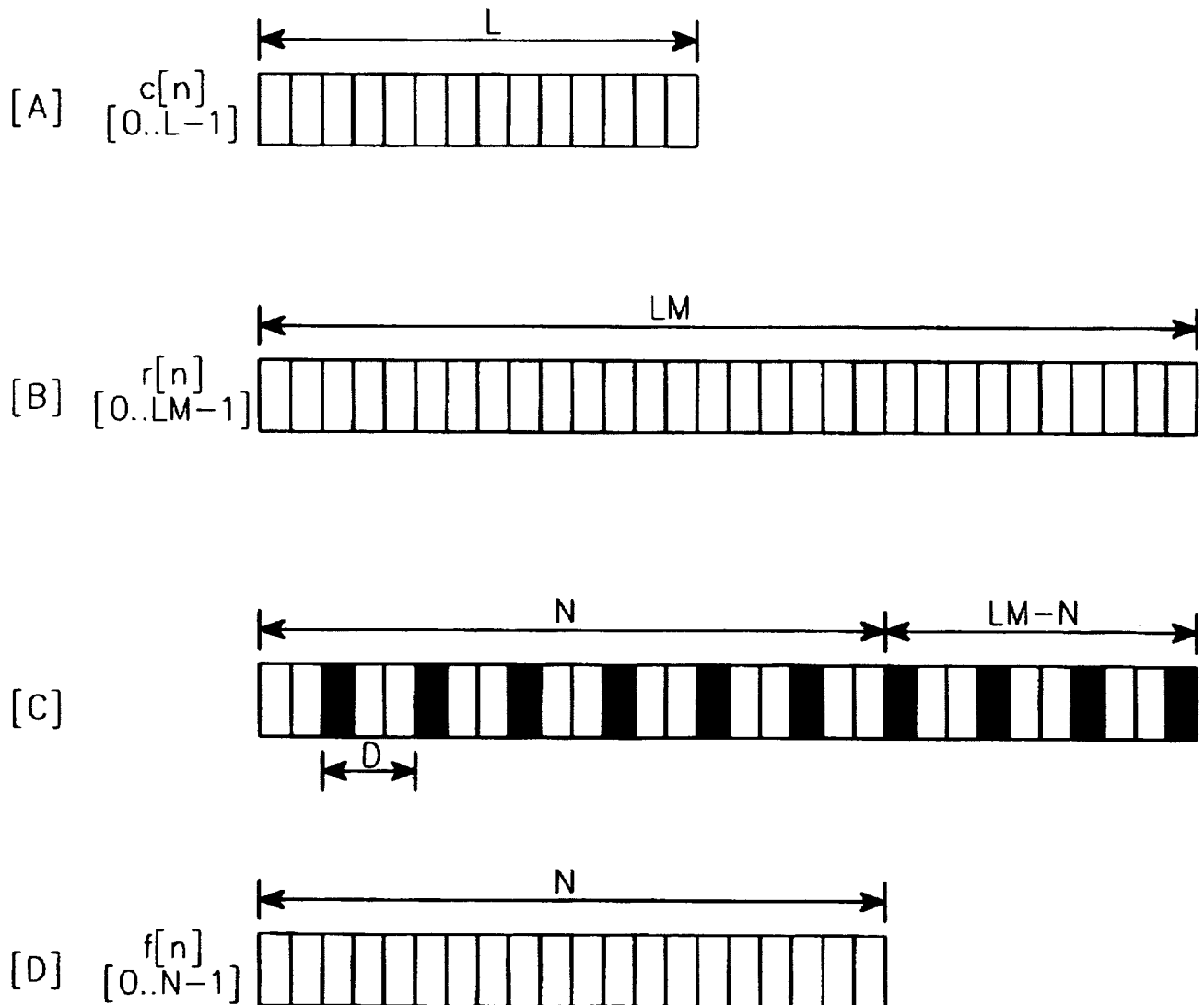
도면 2



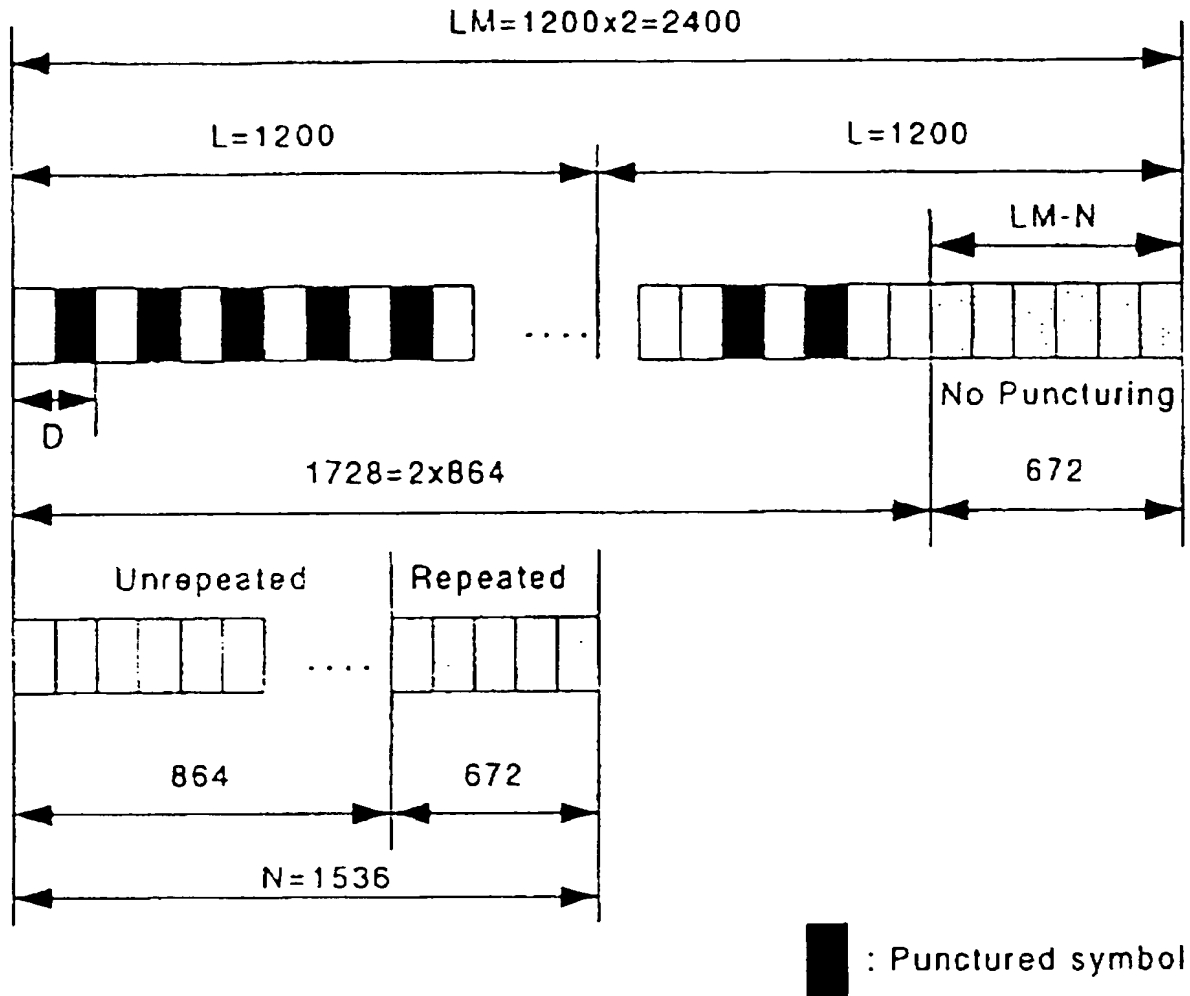
도면 3



도면 4

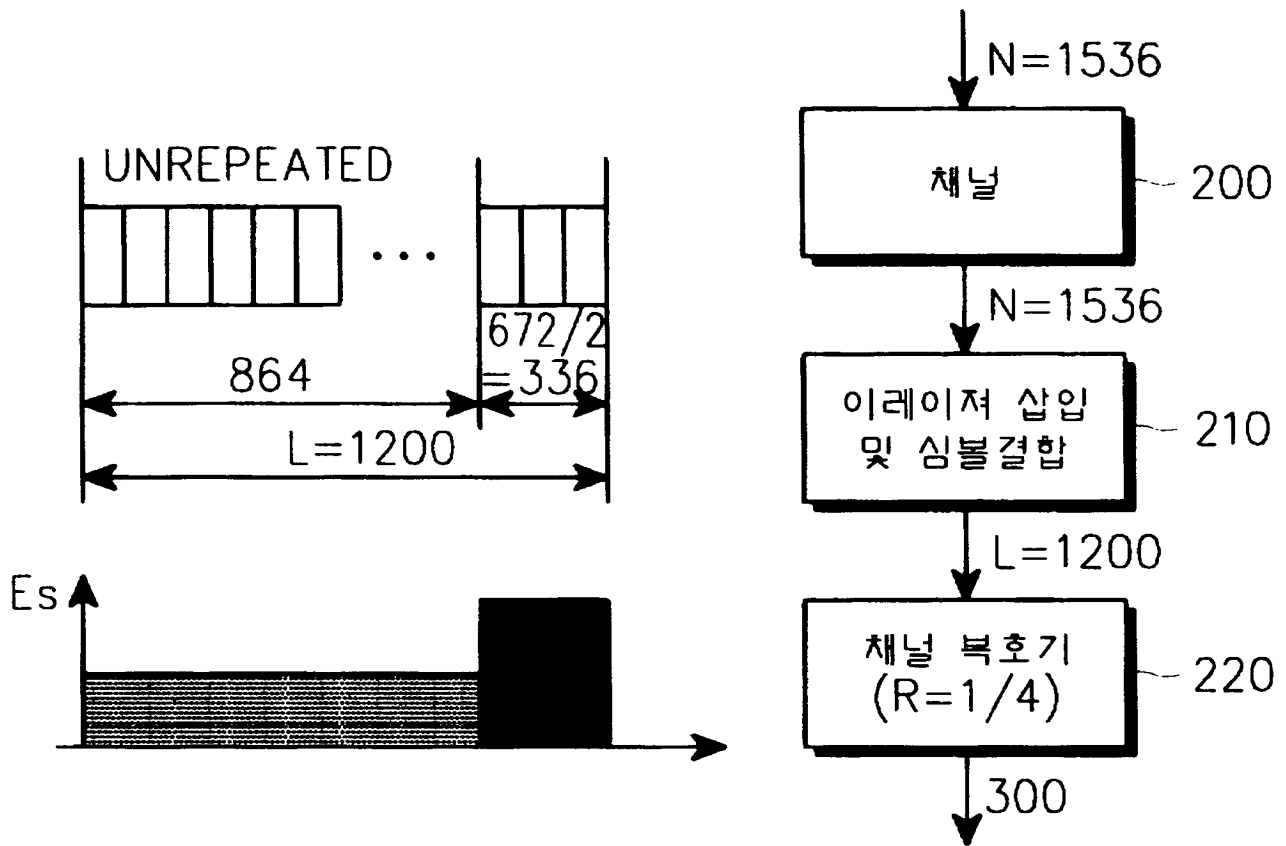


도면 5

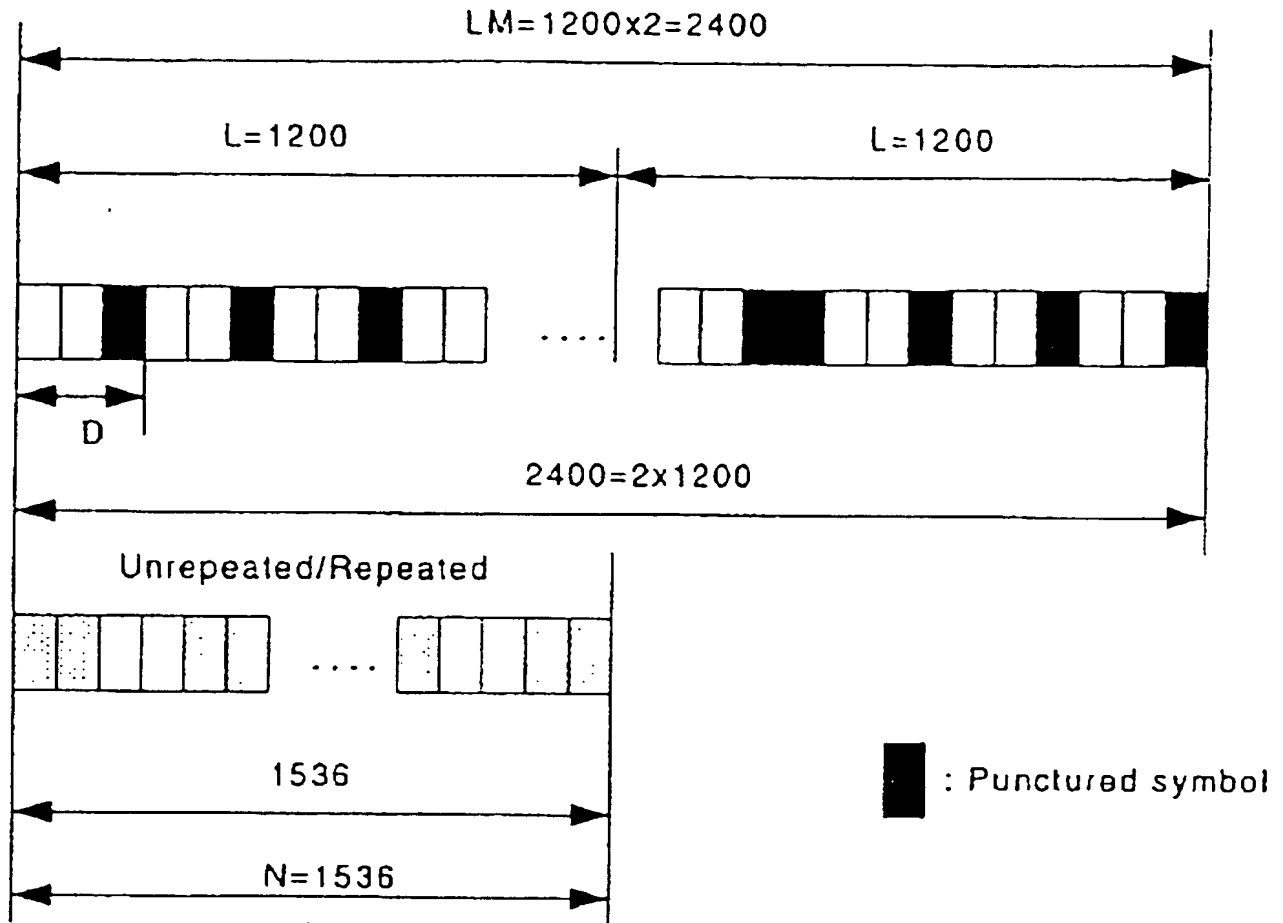


- RC3 19.2kbps 에서 FDRT 로 15kbps 를 전송할 때의 Puncturing pattern.

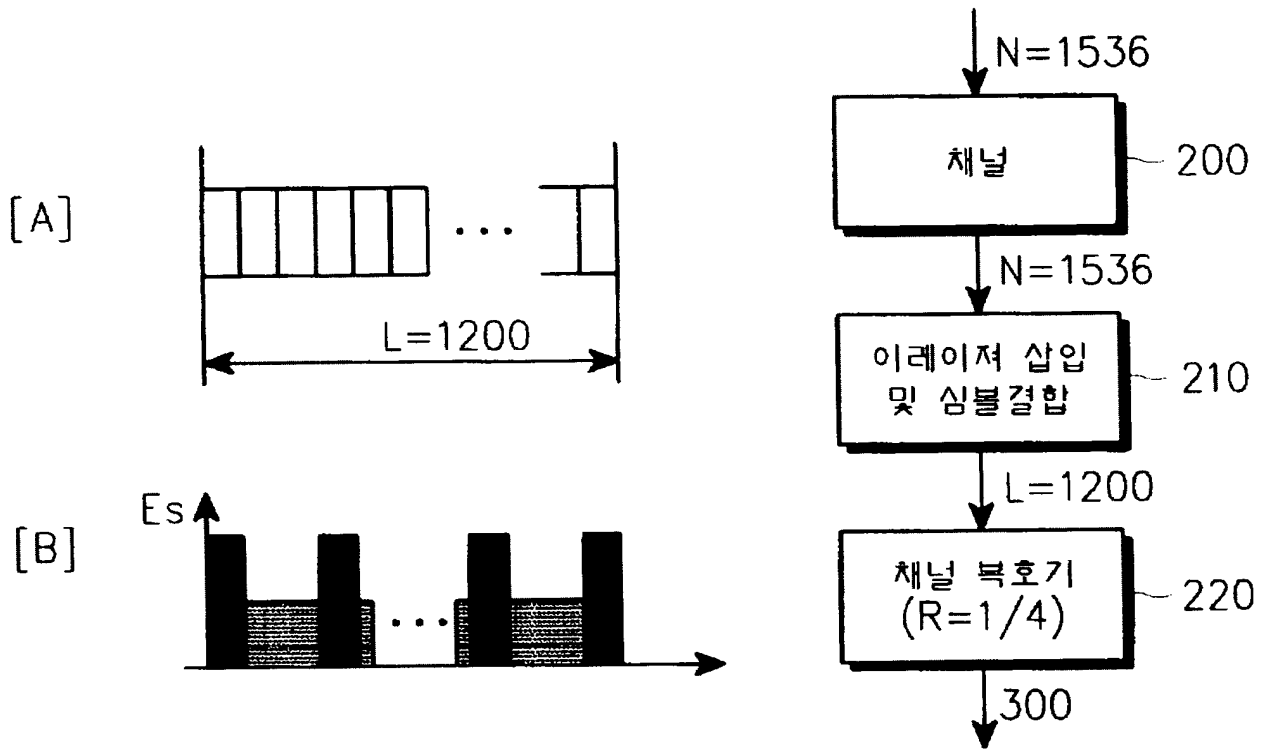
도면 6



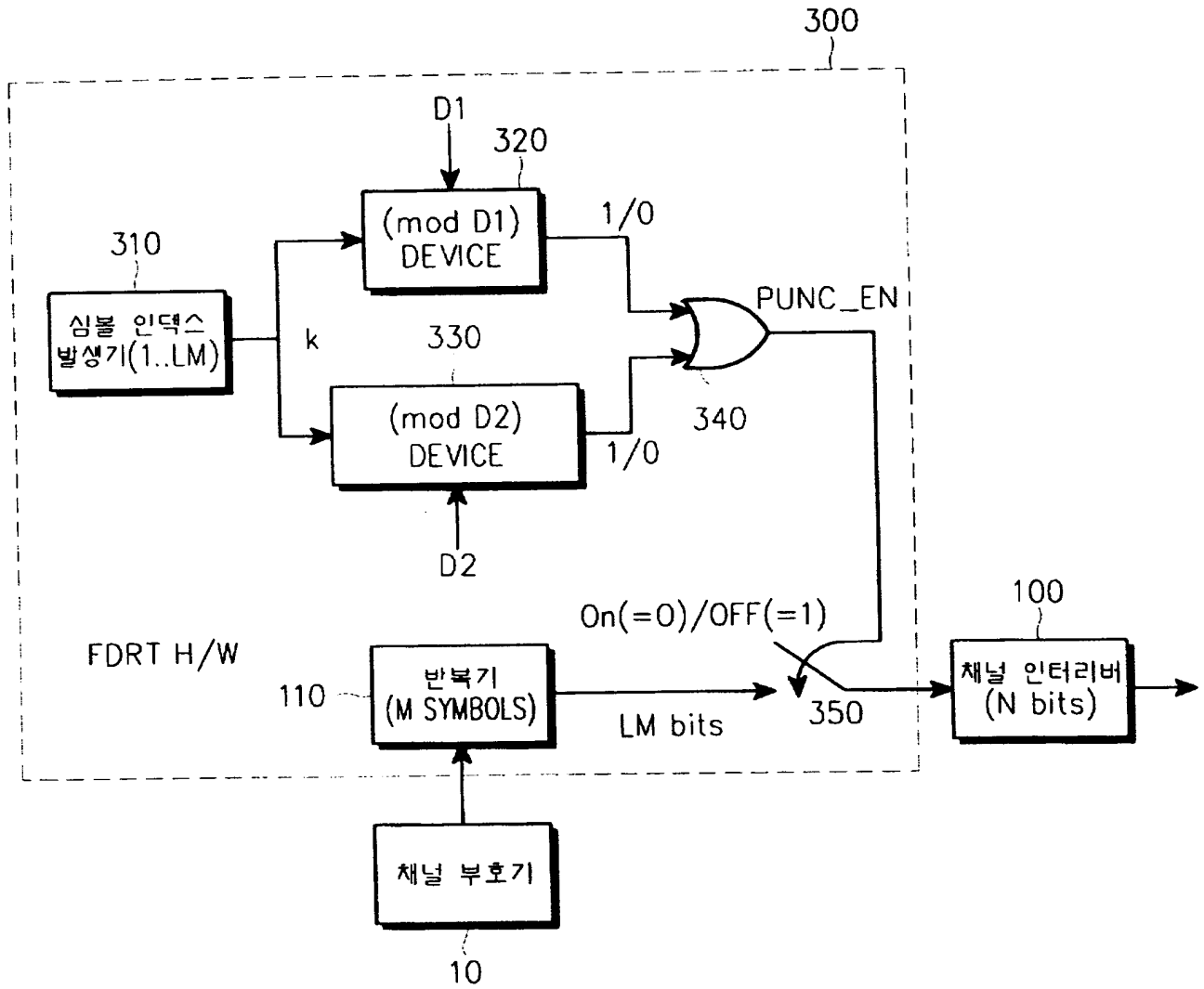
도면 7



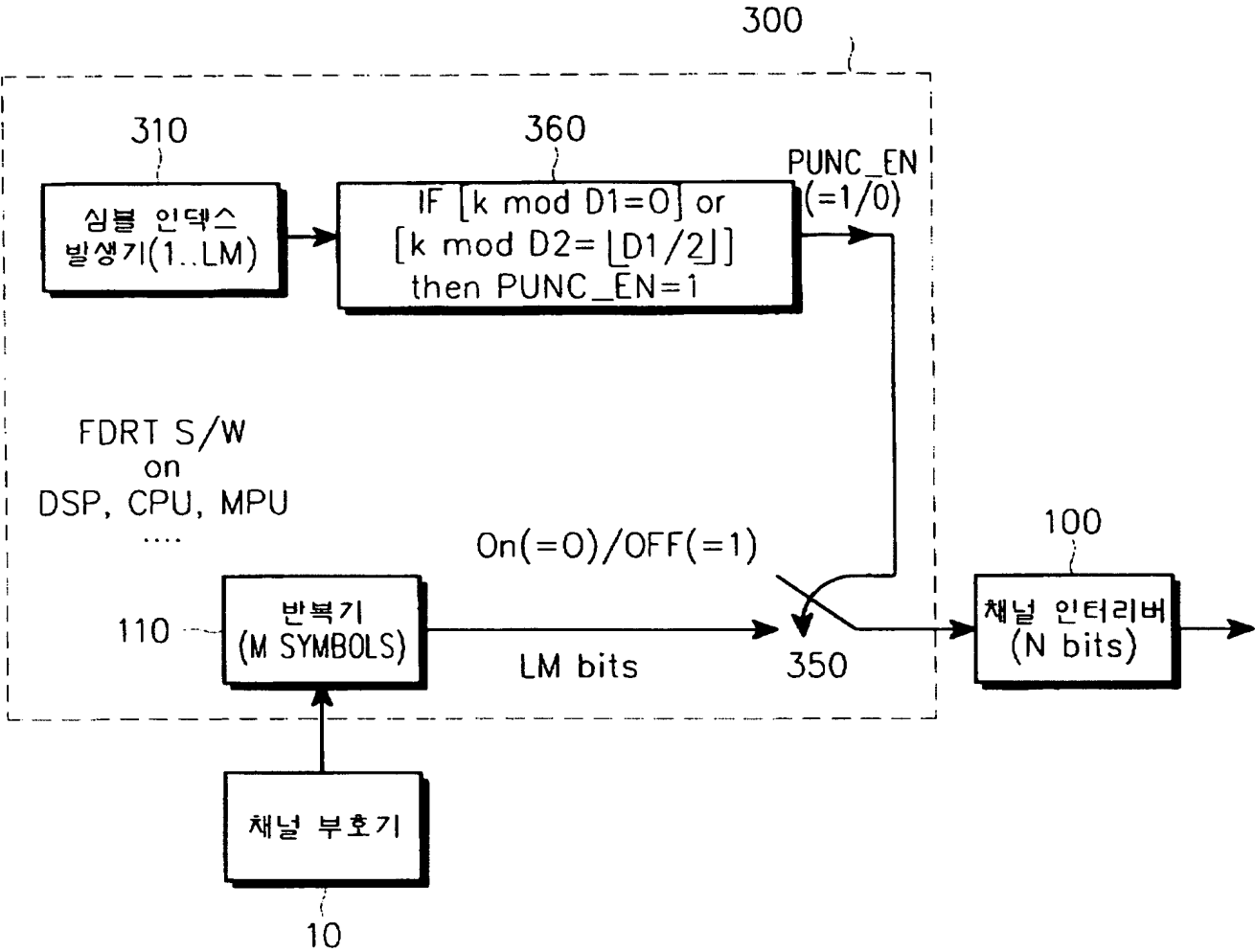
도면 8



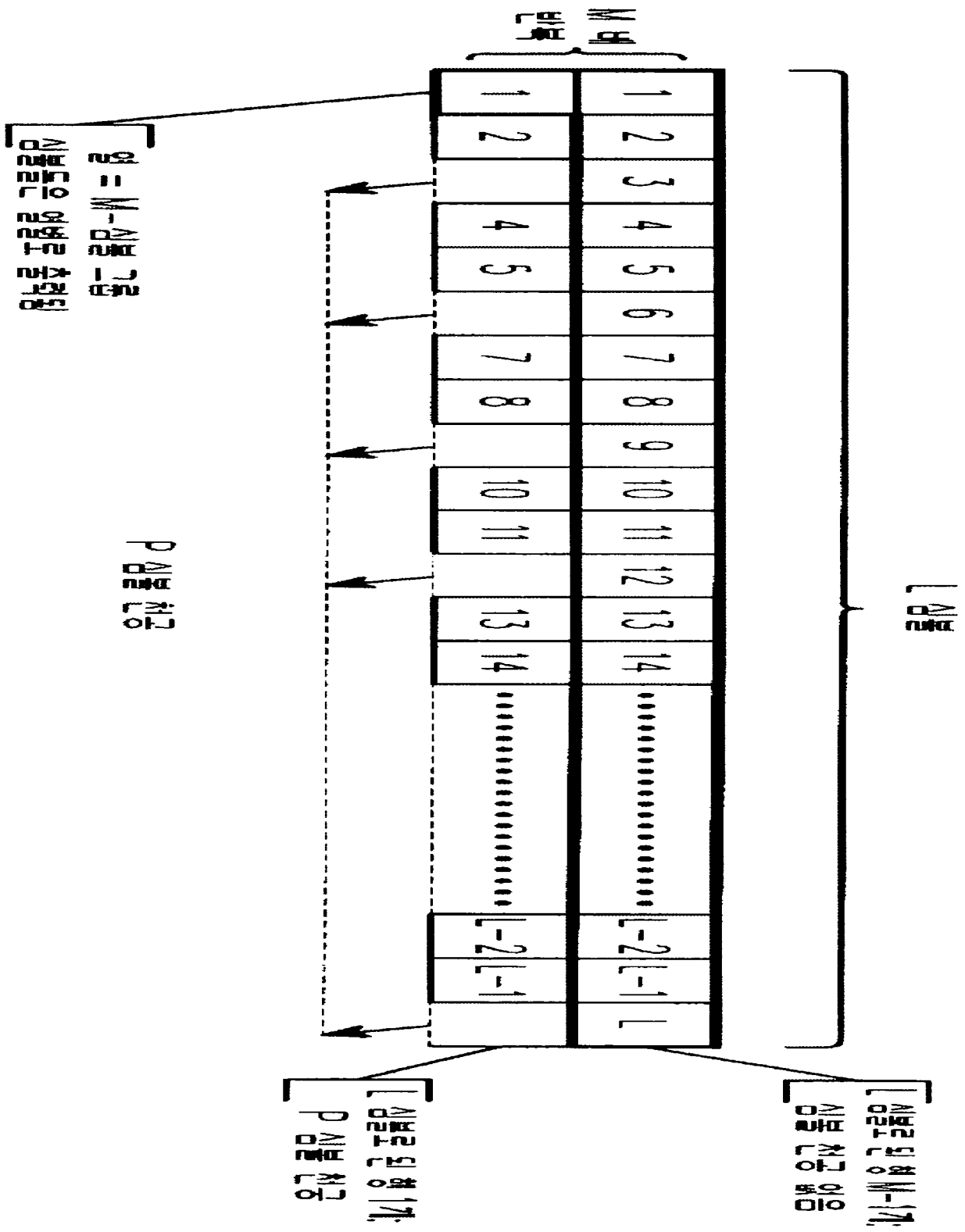
도면 10



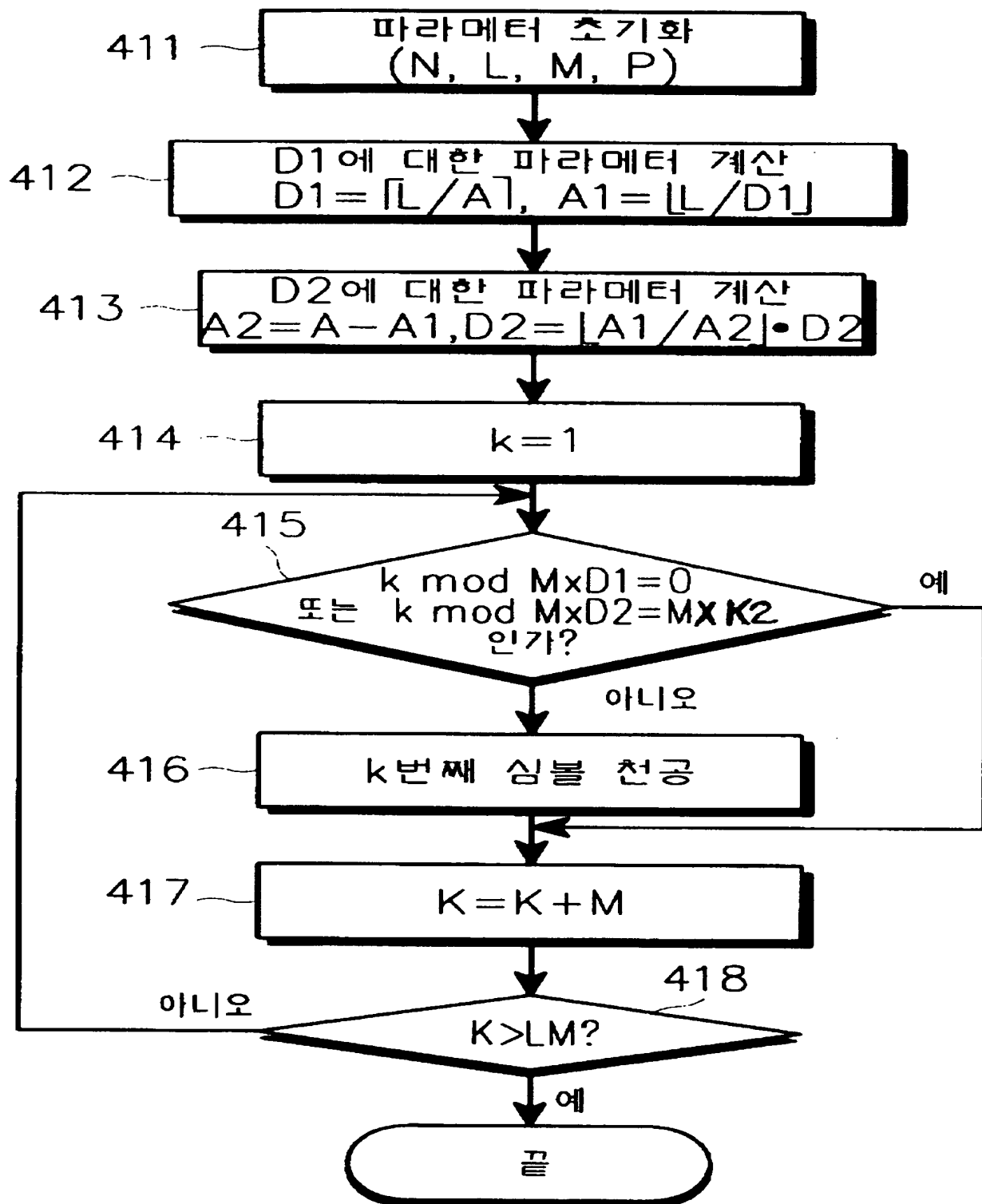
도면 11



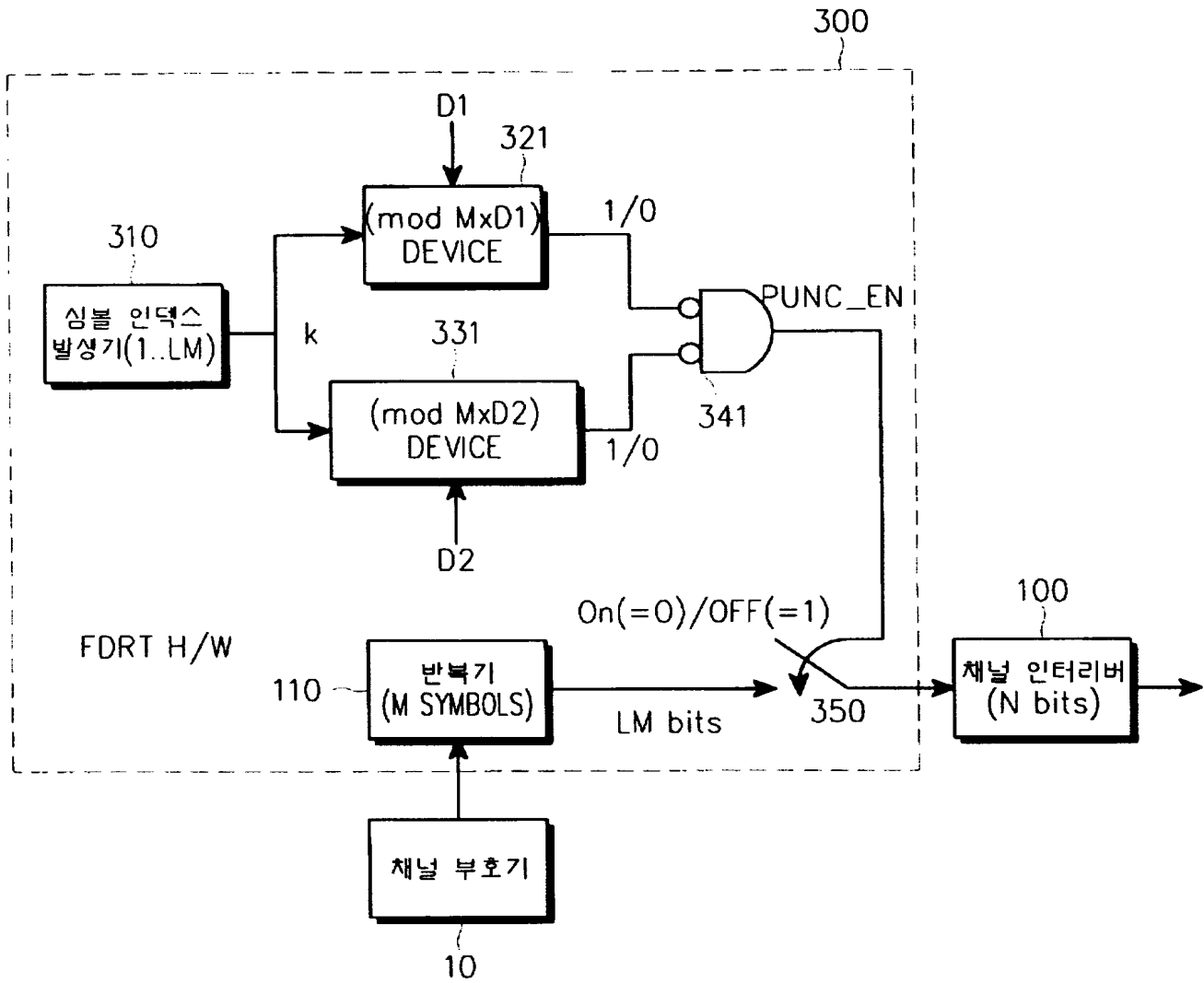
도면 12



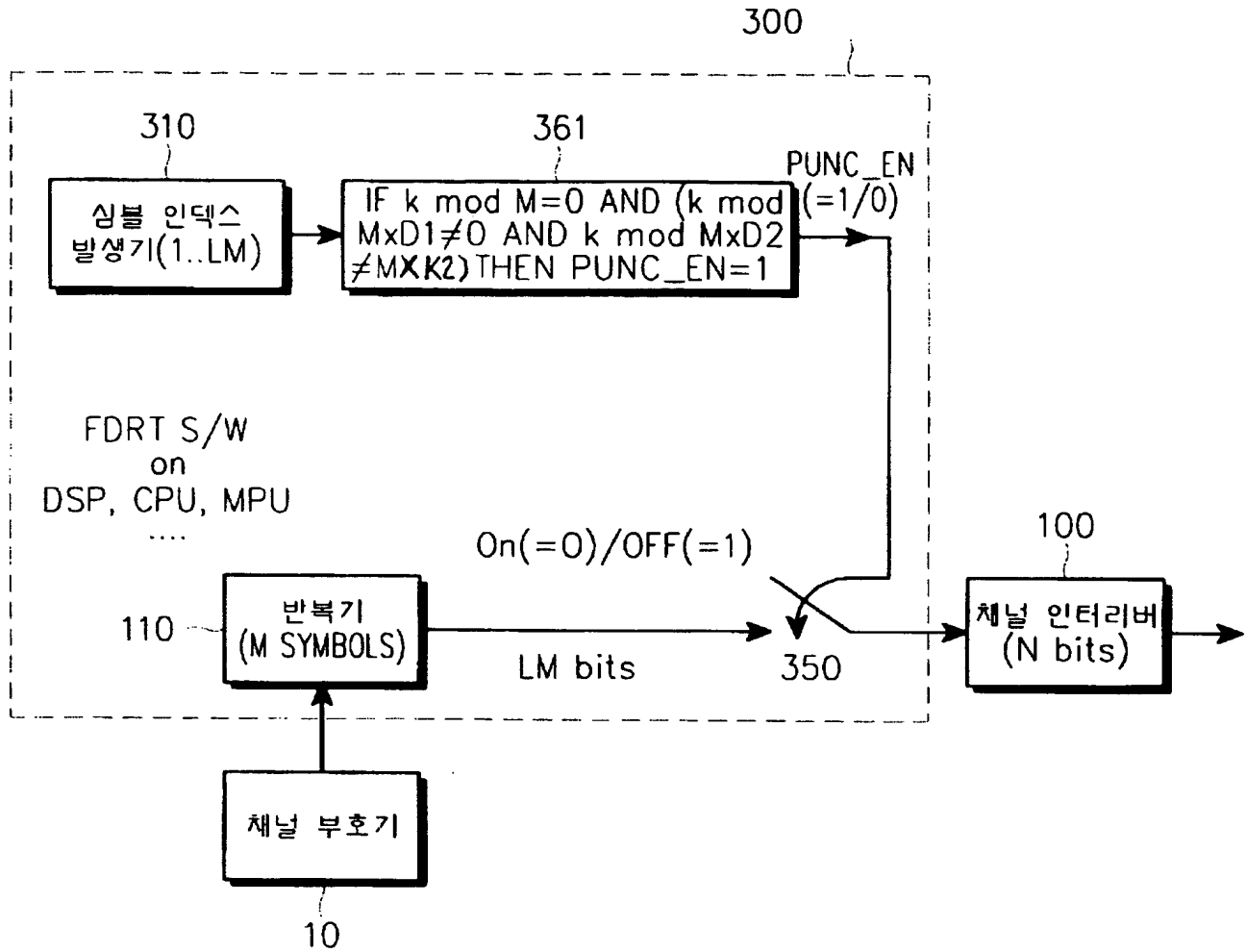
도면 13



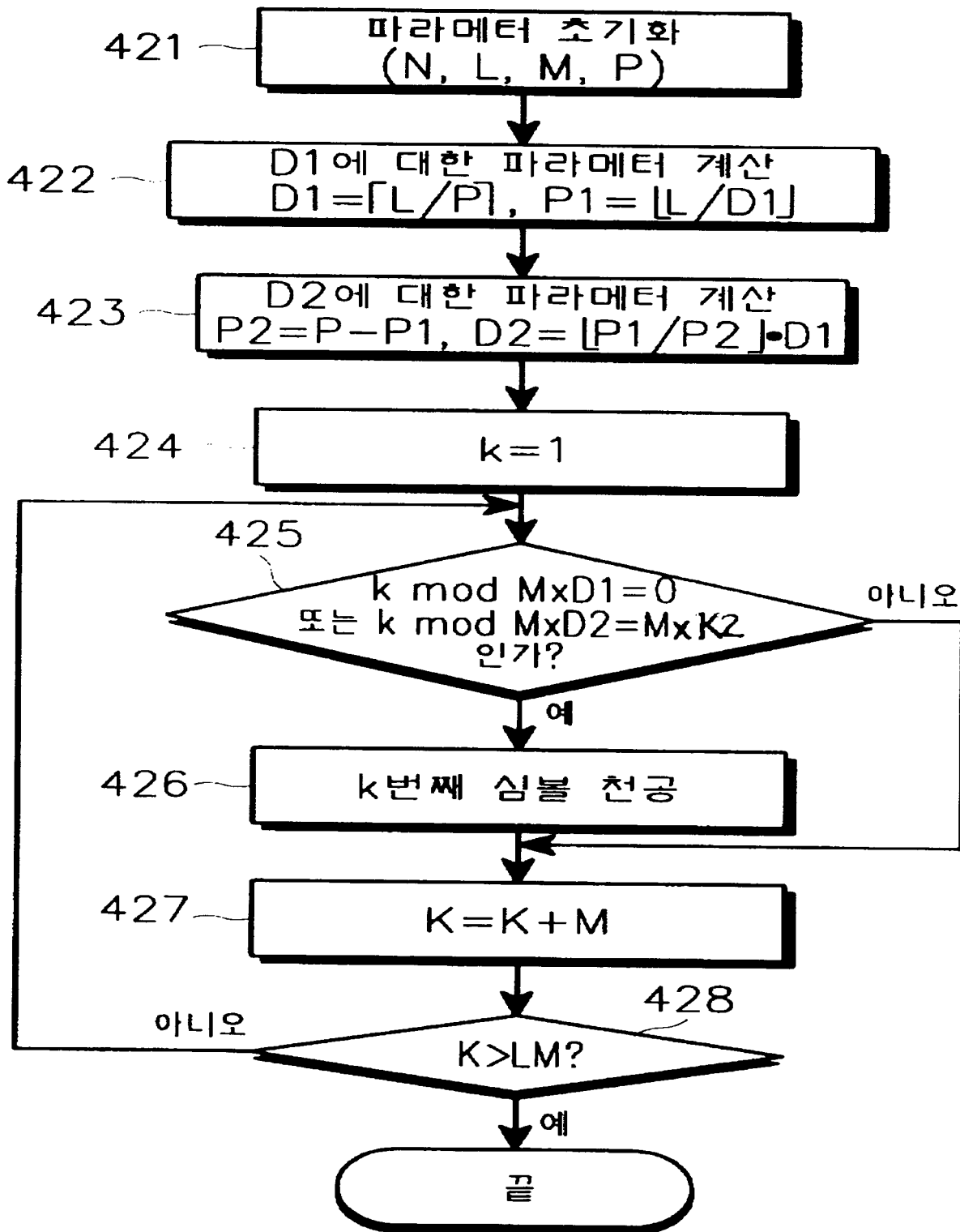
도면 14



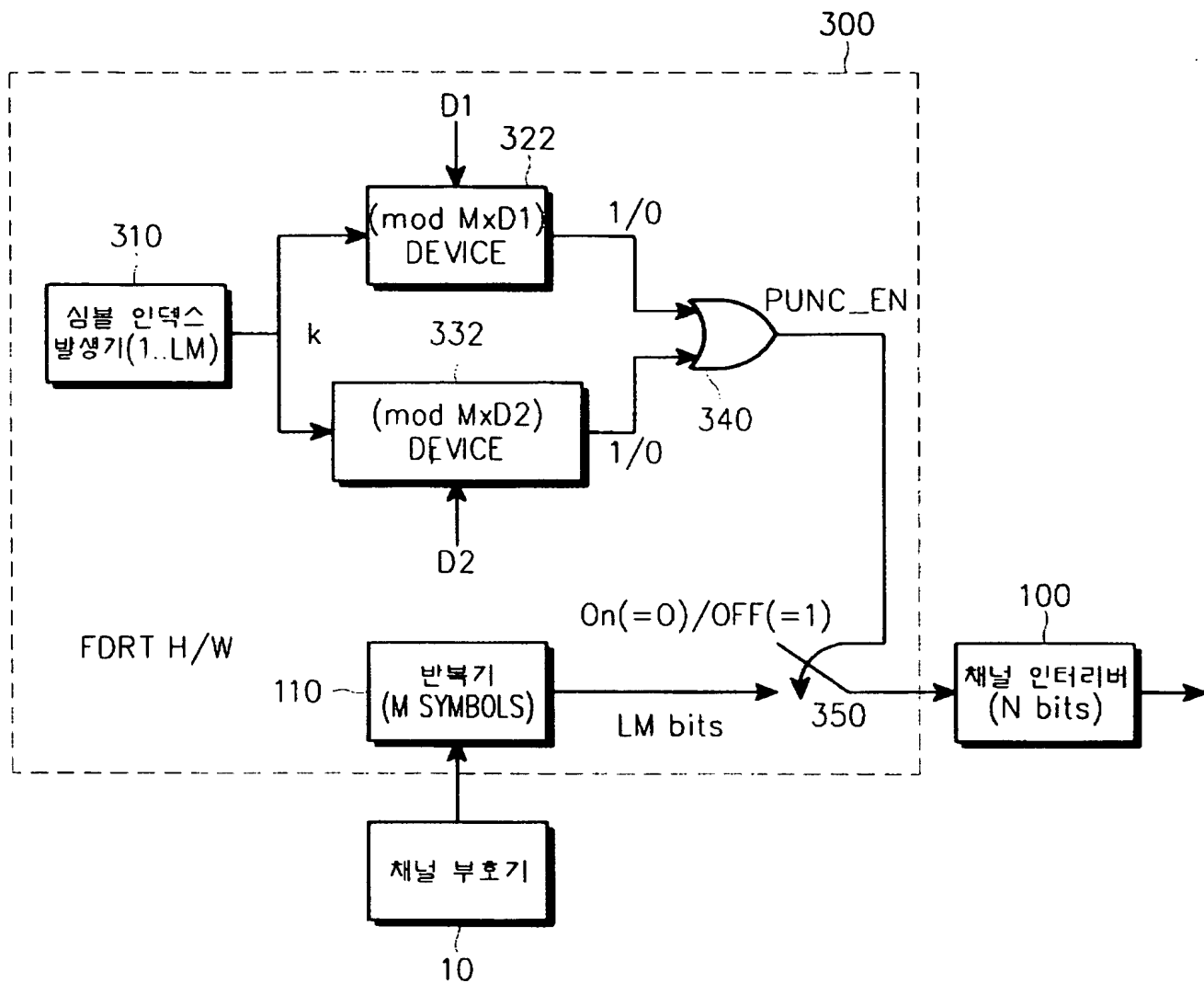
도면 15



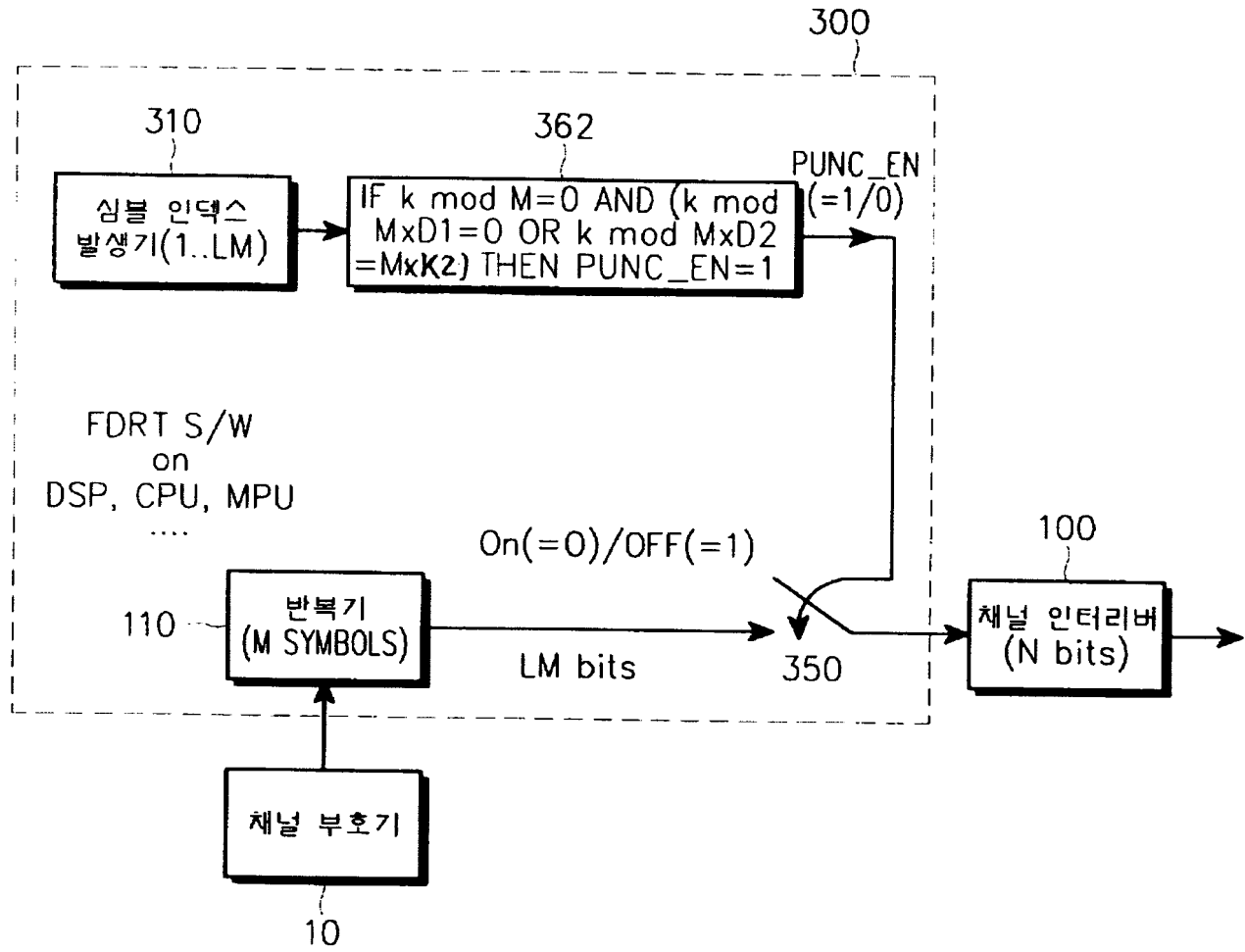
도면 16



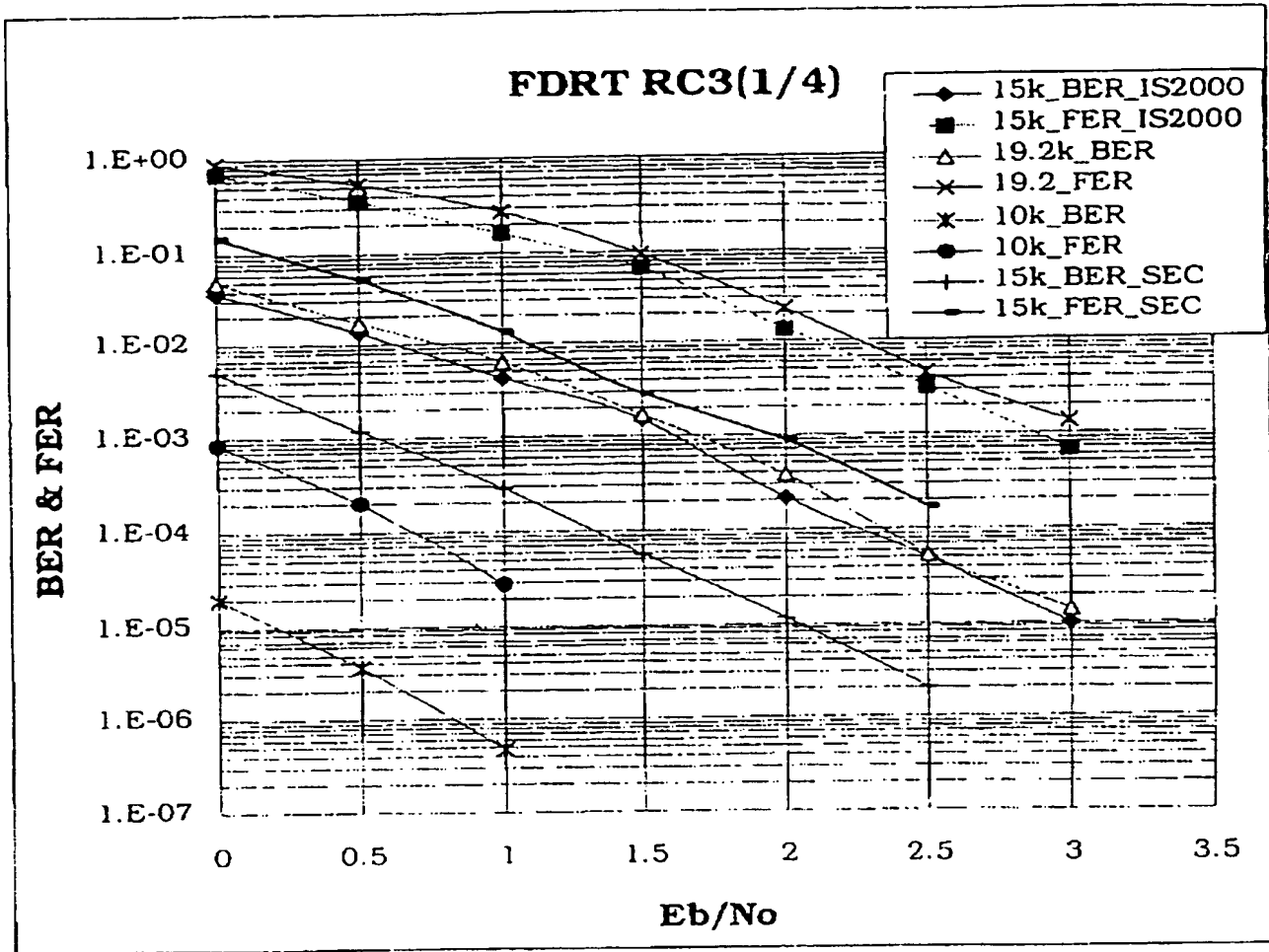
도면 17



도면 18



도면 19



도면 20

